Pas d'erreurs fatales avec le tropical !

Unithé ou café

Xavier Allamigeon

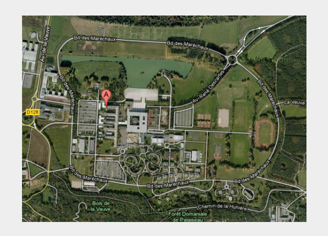
INRIA Saclay – Ile-de-France et CMAP Ecole Polytechnique, équipe MAXPLUS

prenom.nom@inria.fr

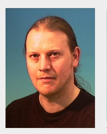
9 mars 2012

L'équipe MAXPLUS

Commune avec le CMAP de l'X, sur le platâl (aile 0 des labos)



- permanents : Stéphane Gaubert (boss), Marianne Akian, Cormac Walsh, XA

- notre assistante : Wallis Filippi
- 1 postdoc (Sepideh Mirrahimi), 6 doctorants (Sylvie, Zheng, Jean-Baptiste, Olivier, Pascal, et Victor)

But de cet exposé

Vous montrer le lien entre :

But de cet exposé

Vous montrer le lien entre :

- la vérification de programmes

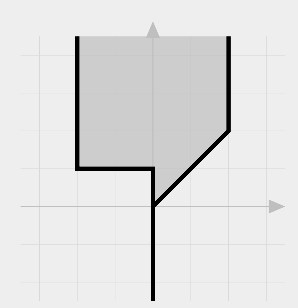
```
x := 0;
tant que x <= 5, faire
  x := x + 2;
// fin de la boucle
y := 1 / x;
```

But de cet exposé

Vous montrer le lien entre :

- la vérification de programmes
- la géométrie tropicale

```
x := 0;
tant que x <= 5, faire
  x := x + 2;
// fin de la boucle
y := 1 / x;
```

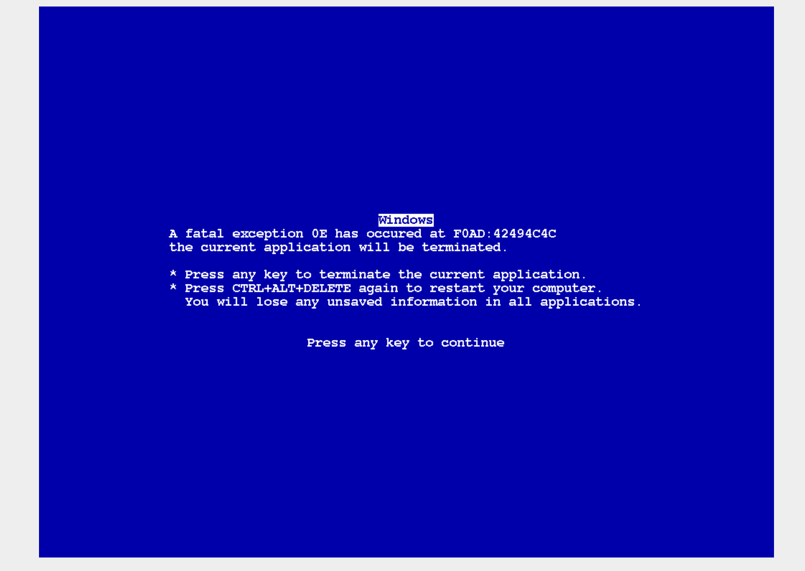


Plan de l'exposé

Plan de l'exposé

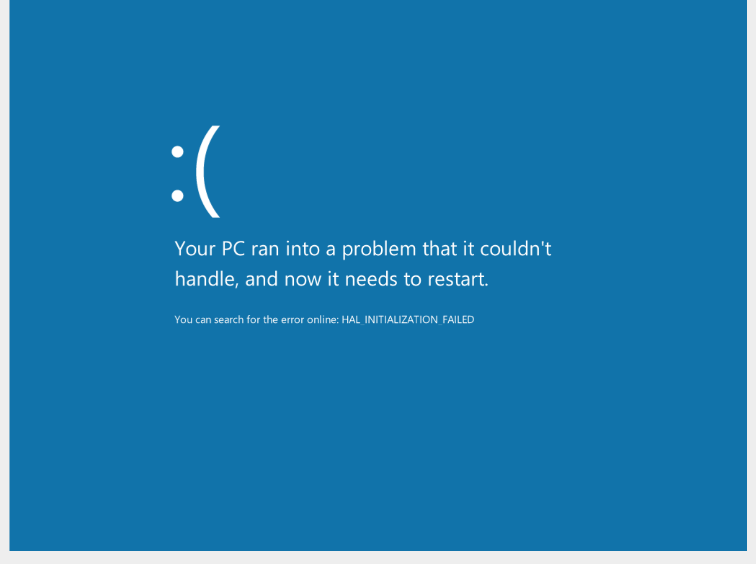
Les bugs font partie de notre quotidien

Leur forme évolue avec le temps :



Les bugs font partie de notre quotidien

Leur forme évolue avec le temps :



Les bugs font partie de notre quotidien

En grand, à Times square :



Les bugs font partie de notre quotidien

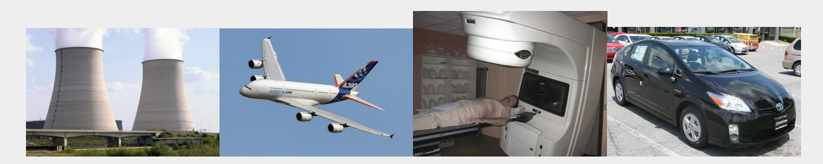
Longue tradition à la SNCF :





On met du logiciel partout

Y compris dans des endroits critiques :

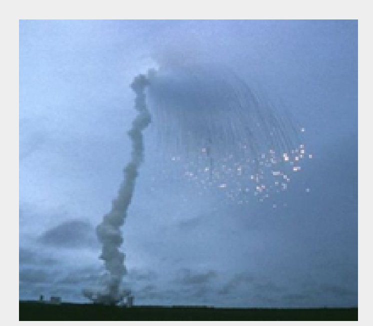


On met du logiciel partout

Y compris dans des endroits critiques :



Ce qui mène parfois à :



La vérification de programmes

Cela consiste à développer des outils d'analyse *statique* de logiciels :

- apportant l'assurance qu'il n'y a pas de bogues
- (presque) entièrement automatiques
- capables de s'attaquer à de très gros codes

La vérification de programmes

Cela consiste à développer des outils d'analyse *statique* de logiciels :

- apportant l'assurance qu'il n'y a pas de bogues
- (presque) entièrement automatiques
- capables de s'attaquer à de très gros codes

Quels genres de bugs ?

- *division par zero* : $1/0$ retourne une erreur
- *dépassement de capacité*

$$15561676378226 * 387326762373690 = -2585705764785308204$$

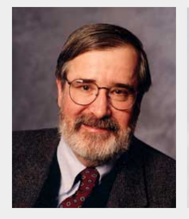
- *etc*

$\Longrightarrow$ les outils de vérification de logiciels doivent en fait s'assurer que certaines *bonnes* propriétés sont satisfaites.

La vérification de programmes (2)

Trois grands paroisses :

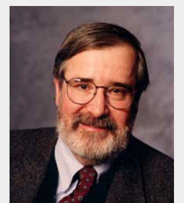
- l'approche par model checking (Clarke, Sifakis, Prix Turing 2007)




La vérification de programmes (2)

Trois grands paroisses :

- l'approche par model checking (Clarke, Sifakis, Prix Turing 2007)

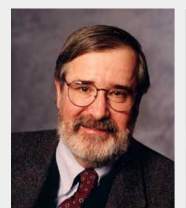



- par assistant de preuve, par exemple Why (PROVAL)

La vérification de programmes (2)

Trois grands paroisses :

- l'approche par model checking (Clarke, Sifakis, Prix Turing 2007)




- par assistant de preuve, par exemple Why (PROVAL)
- par interprétation abstraite (Cousot & Cousot)




La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
x := 0;
tant que x est plus petit que 5, faire
  x := x + 2;
// fin de la boucle
y := 1 / x;
```

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
1 :  x := 0;
2 :  tant que x est plus petit que 5, faire
3 :    x := x + 2;
4 :  // fin de la boucle
5 :  y := 1 / x;
```

- à la ligne 1, x vaut 0

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
1 :  x := 0;
2 :  tant que x est plus petit que 5, faire
3 :    x := x + 2;
4 :  // fin de la boucle
5 :  y := 1 / x;
```

- à la ligne 1, x vaut 0
- comme x est plus petit que 5, on rentre dans la boucle

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
1 :  x := 0;
2 :  tant que x est plus petit que 5, faire
3 :    x := x + 2;
4 :  // fin de la boucle
5 :  y := 1 / x;
```

- à la ligne 1, x vaut 0
- comme x est plus petit que 5, on rentre dans la boucle
- à la ligne 3, x vaut 2,

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
1 :  x := 0;
2 :  tant que x est plus petit que 5, faire
3 :    x := x + 2;
4 :  // fin de la boucle
5 :  y := 1 / x;
```

- à la ligne 1, x vaut 0
- comme x est plus petit que 5, on rentre dans la boucle
- à la ligne 3, x vaut 2, puis 4,

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
1 :  x := 0;
2 :  tant que x est plus petit que 5, faire
3 :    x := x + 2;
4 :  // fin de la boucle
5 :  y := 1 / x;
```

- à la ligne 1, x vaut 0
- comme x est plus petit que 5, on rentre dans la boucle
- à la ligne 3, x vaut 2, puis 4, puis 6

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
1 :  x := 0;
2 :  tant que x est plus petit que 5, faire
3 :    x := x + 2;
4 :  // fin de la boucle
5 :  y := 1 / x;
```

- à la ligne 1, x vaut 0
- comme x est plus petit que 5, on rentre dans la boucle
- à la ligne 3, x vaut 2, puis 4, puis 6
- on sort de la boucle, on saute à la ligne 5

La vérification de programmes par interprétation abstraite

La sémantique d'un programme décrit l'ensemble de *tous* les comportements possibles du programme.

Exemple très simple :

```
1 :  x := 0;
2 :  tant que x est plus petit que 5, faire
3 :    x := x + 2;
4 :  // fin de la boucle
5 :  y := 1 / x;
```

- à la ligne 1, x vaut 0
- comme x est plus petit que 5, on rentre dans la boucle
- à la ligne 3, x vaut 2, puis 4, puis 6
- on sort de la boucle, on saute à la ligne 5
- à la ligne 5, x vaut 6, et y vaut 1/6

La vérification de programmes par interprétation abstraite (2)

- la sémantique d'un programme n'est pas calculable en général



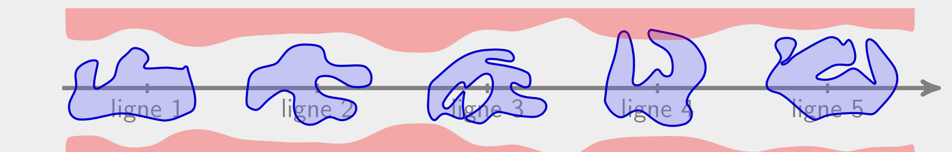
La vérification de programmes par interprétation abstraite (2)

- la sémantique d'un programme n'est pas calculable en général



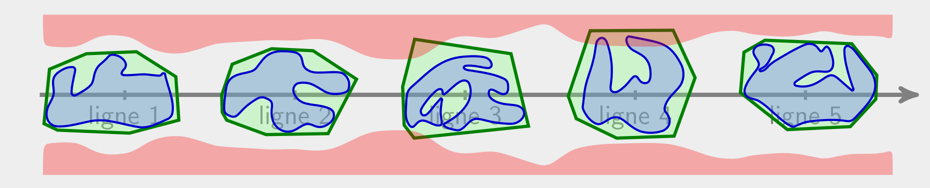
La vérification de programmes par interprétation abstraite (2)

- la sémantique d'un programme n'est pas calculable en général
- et pourtant on en aurait besoin pour déterminer s'il y a des bugs



La vérification de programmes par interprétation abstraite (2)

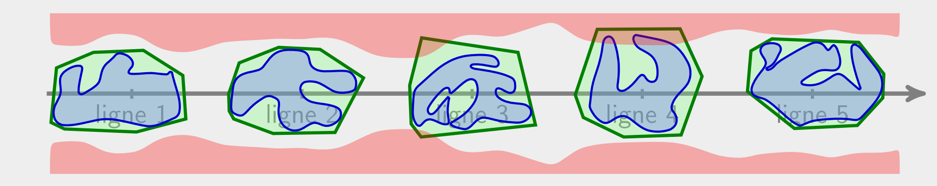
- la sémantique d'un programme n'est pas calculable en général
- et pourtant on en aurait besoin pour déterminer s'il y a des bugs



Principe de l'IA : on calcule une sur-approximation de la sémantique

La vérification de programmes par interprétation abstraite (2)

- la sémantique d'un programme n'est pas calculable en général
- et pourtant on en aurait besoin pour déterminer s'il y a des bugs

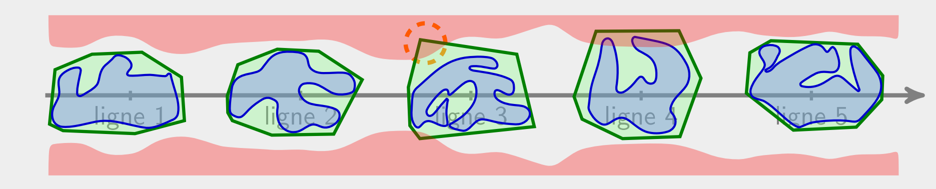


Principe de l'IA : on calcule une sur-approximation de la sémantique

- on ne peut rater aucun bug

La vérification de programmes par interprétation abstraite (2)

- la sémantique d'un programme n'est pas calculable en général
- et pourtant on en aurait besoin pour déterminer s'il y a des bugs



Principe de l'IA : on calcule une sur-approximation de la sémantique

- on ne peut rater aucun bug
- mais si on n'est pas assez précis, on peut croire qu'il y a un bug (alors qu'en fait non) → fausse alarme !

L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
$\longrightarrow$ on encadre chaque variable entre deux constantes

L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
$\longrightarrow$ on encadre chaque variable entre deux constantes

En pratique :

```
si [condition aléatoire] alors
  x := 5, y := 0;
sinon
  x := 0, y := 5;
// fin de la conditionnelle

si y >= 3 alors
  x := x+1;
```

L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
⟶ on encadre chaque variable entre deux constantes

En pratique :

- ligne 2 : $x = 5$, $y = 0$,

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
⟶ on encadre chaque variable entre deux constantes

En pratique :

```
1 : si [condition aléatoire] alors
2 :   x := 5, y := 0;
3 : sinon
4 :   x := 0, y := 5;
5 : // fin de la conditionnelle

6 : si y >= 3 alors
7 :   x := x+1;
```

- ligne 2 : $x = 5$, $y = 0$,
- ligne 4 : $x = 0$, $y = 5$,

L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
$\longrightarrow$ on encadre chaque variable entre deux constantes

En pratique :

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

- ligne 2 : $x = 5$, $y = 0$,
- ligne 4 : $x = 0$, $y = 5$,
- ligne 5 : l'un des deux cas au-dessus ! donc :

$$0 \leq x \leq 5 \quad 0 \leq y \leq 5$$

L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
⟶ on encadre chaque variable entre deux constantes

En pratique :

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

- ligne 2 : $x = 5$, $y = 0$,
- ligne 4 : $x = 0$, $y = 5$,
- ligne 5 : l'un des deux cas au-dessus ! donc :

$$0 \leq x \leq 5 \quad 0 \leq y \leq 5$$

- juste après la ligne 6 :

$$0 \leq x \leq 5 \quad 3 \leq y \leq 5$$

L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
$\longrightarrow$ on encadre chaque variable entre deux constantes

En pratique :

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

- ligne 2 : $x = 5$, $y = 0$,
- ligne 4 : $x = 0$, $y = 5$,
- ligne 5 : l'un des deux cas au-dessus ! donc :

  $0 \leq x \leq 5 \quad 0 \leq y \leq 5$

- juste après la ligne 6 :

  $0 \leq x \leq 5 \quad 3 \leq y \leq 5$

- à la ligne 7 :

  $1 \leq x \leq 6 \quad 3 \leq y \leq 5$

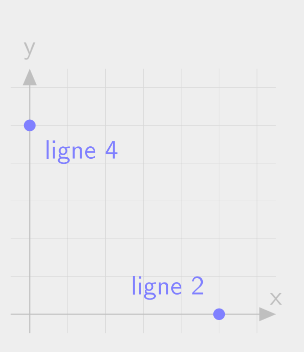
L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
⟶ on encadre chaque variable entre deux constantes

En pratique :

```
1 : si [condition aléatoire] alors
2 :   x := 5, y := 0;
3 : sinon
4 :   x := 0, y := 5;
5 : // fin de la conditionnelle

6 : si y >= 3 alors
7 :   x := x+1;
```



L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
$\longrightarrow$ on encadre chaque variable entre deux constantes

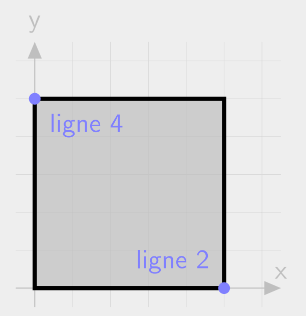
En pratique :

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

A la ligne 5 :

$0 \leq x \leq 5 \quad 0 \leq y \leq 5$



L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
$\longrightarrow$ on encadre chaque variable entre deux constantes

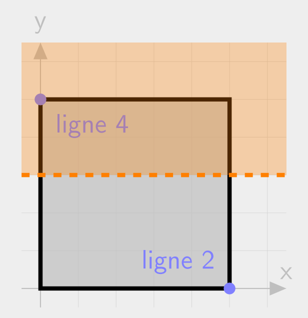
En pratique :

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

Juste après la ligne 6 :

$0 \leq x \leq 5 \quad 3 \leq y \leq 5$



L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
⟶ on encadre chaque variable entre deux constantes

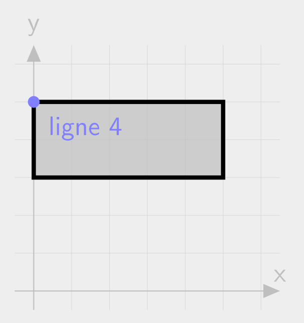
En pratique :

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

Juste après la ligne 6 :

$0 \leq x \leq 5 \quad 3 \leq y \leq 5$



L'interprétation abstraite : crash course

Exemple de sur-approximations : les intervalles
$\longrightarrow$ on encadre chaque variable entre deux constantes

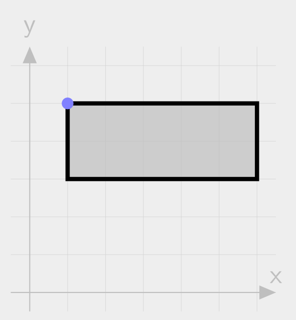
En pratique :

```
1 :  si [condition aléatoire] alors
2 :    x := 5, y := 0;
3 :  sinon
4 :    x := 0, y := 5;
5 :  // fin de la conditionnelle

6 :  si y >= 3 alors
7 :    x := x+1;
```

A la ligne 7 :

$1 \leq x \leq 6 \quad 3 \leq y \leq 5$



L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

- intervals $1 \leq \mathtt{x} \leq 3$ (Cousot & Cousot)

L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

- intervals $1 \leq \mathtt{x} \leq 3$ (Cousot & Cousot)
- zones $\mathtt{y} - \mathtt{x} \geq 1$ (Miné)

L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

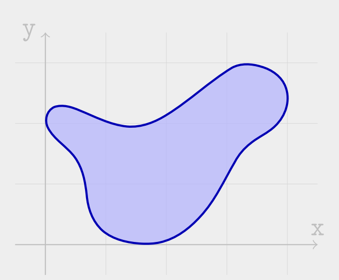
- intervals $1 \leq \mathtt{x} \leq 3$ (Cousot & Cousot)
- zones $\mathtt{y} - \mathtt{x} \geq 1$ (Miné)
- polyèdres convexes $2\mathtt{x} + 5\mathtt{y} \geq -3$ (Cousot & Halbwachs)

L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

- intervals $1 \leq x \leq 3$ (Cousot & Cousot)
- zones $y - x \geq 1$ (Miné)
- polyèdres convexes $2x + 5y \geq -3$ (Cousot & Halbwachs)

... qui offrent différents niveaux de précision

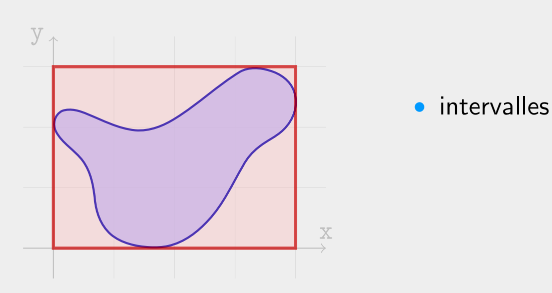


L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

- intervals $1 \leq x \leq 3$ (Cousot & Cousot)
- zones $y - x \geq 1$ (Miné)
- polyèdres convexes $2x + 5y \geq -3$ (Cousot & Halbwachs)

... qui offrent différents niveaux de précision



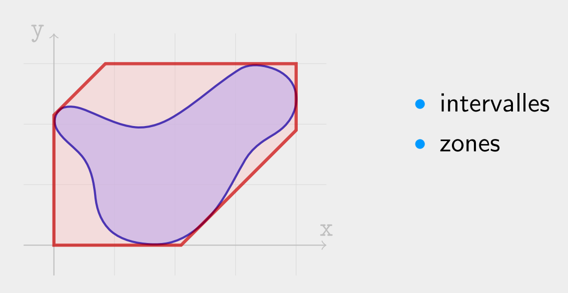
- intervalles

L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

- intervals $1 \leq x \leq 3$ (Cousot & Cousot)
- zones $y - x \geq 1$ (Miné)
- polyèdres convexes $2x + 5y \geq -3$ (Cousot & Halbwachs)

... qui offrent différents niveaux de précision



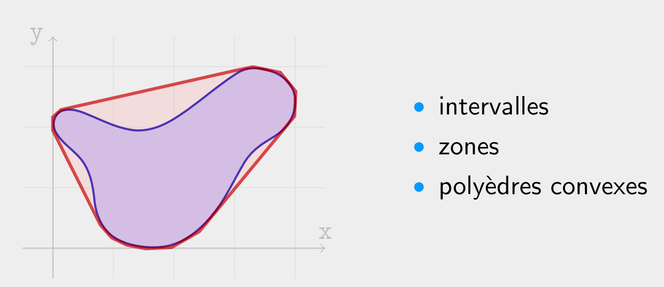
- intervalles
- zones

L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

- intervals $1 \leq x \leq 3$ (Cousot & Cousot)
- zones $y - x \geq 1$ (Miné)
- polyèdres convexes $2x + 5y \geq -3$ (Cousot & Halbwachs)

... qui offrent différents niveaux de précision



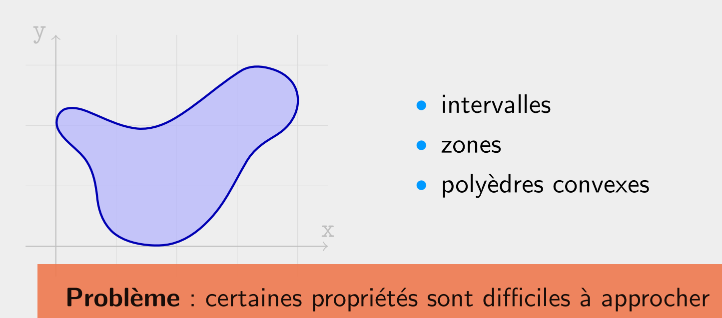
- intervalles
- zones
- polyèdres convexes

L'interprétation abstraite : crash course (2)

Il y a plusieurs moyens d'approximer, par exemple :

- intervals $1 \leq \mathtt{x} \leq 3$ (Cousot & Cousot)
- zones $\mathtt{y} - \mathtt{x} \geq 1$ (Miné)
- polyèdres convexes $2\mathtt{x} + 5\mathtt{y} \geq -3$ (Cousot & Halbwachs)

. . . qui offrent différents niveaux de précision



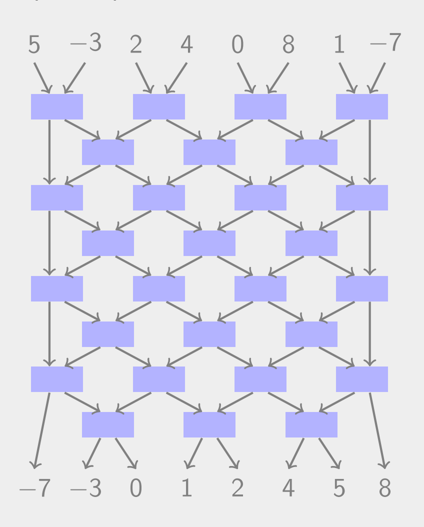
- intervalles
- zones
- polyèdres convexes

Problème : certaines propriétés sont difficiles à approcher

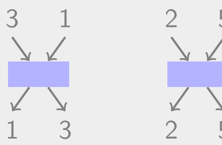
Plan de l'exposé

Le B.A.-BA : les algorithmes de tris

Le tri pair-impair :

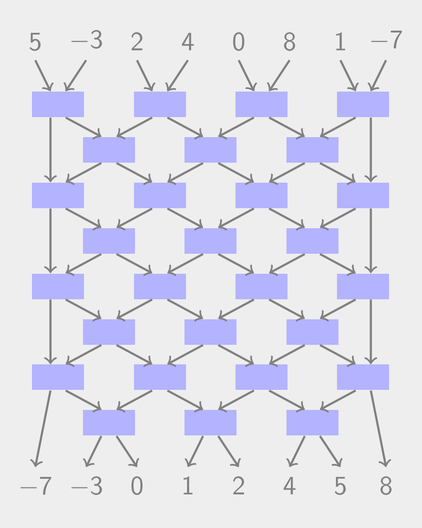


Blocs élémentaires :

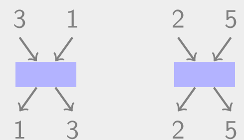


Le B.A.-BA : les algorithmes de tris

Le tri pair-impair :



Blocs élémentaires :



A la sortie,

- l'élément le plus à gauche est le minimum
- l'élément le plus à droite est le maximum

des éléments données en entrée.

Synthèse de plusieurs capteurs

Plusieurs capteurs pour mesurer la décélération :

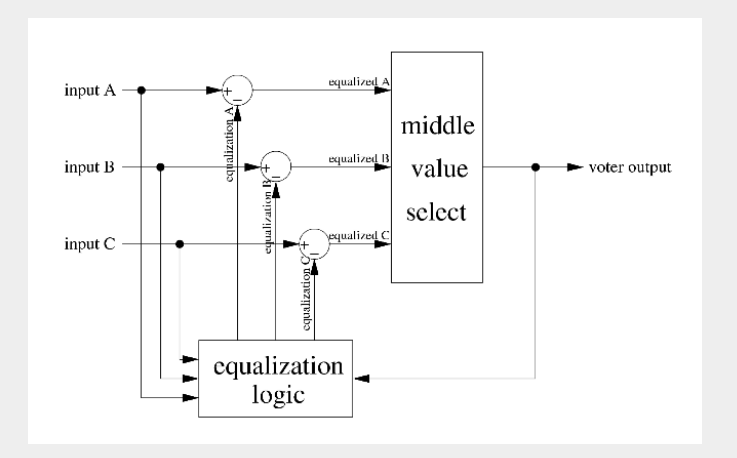


L'airbag se déclenche dès qu'un des capteurs mesure une trop grande décélération :

$$\max(\mathtt{d}_1, \mathtt{d}_2, \mathtt{d}_3) \geq \mathtt{dmax}$$

Synthèse de plusieurs capteurs (2)

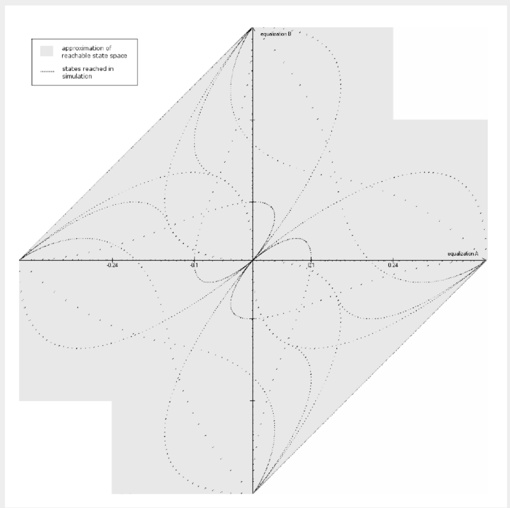
En réalité, c'est plus compliqué ⟶ triplex sensor voter



(Dierkes, Cofer, Ervin, Miller, TAPAS 2010)

Synthèse de plusieurs capteurs (2)

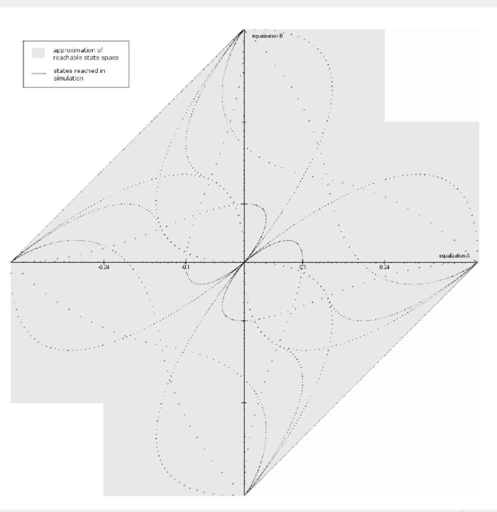
En réalité, c'est plus compliqué ⟶ triplex sensor voter



(Dierkes, Cofer, Ervin, Miller, TAPAS 2010)

Synthèse de plusieurs capteurs (2)

En réalité, c'est plus compliqué ⟶ triplex sensor voter



(Dierkes, Cofer, Ervin, Miller, TAPAS 2010)

Propriétés à verifier :

$$\min(\mathtt{valA}, \mathtt{valB}) \leq 0.24$$
$$\min(\mathtt{valB}, \mathtt{valC}) \leq 0.24$$
$$\min(\mathtt{valA}, \mathtt{valC}) \leq 0.24$$

$$\max(\mathtt{valA}, \mathtt{valB}) \geq -0.24$$
$$\max(\mathtt{valB}, \mathtt{valC}) \geq -0.24$$
$$\max(\mathtt{valA}, \mathtt{valC}) \geq -0.24$$

Manipulations de mémoire

C'est l'utilisation, en programmation, de

- tableaux, matrices
- chaînes de caractères (`"Hello World!"`)
- listes, arbres, *etc*

$\Longrightarrow$ largement utilisé dans les langages de programmation moderne : C, C++, Java, *etc*

Manipulations de mémoire

C'est l'utilisation, en programmation, de

- tableaux, matrices
- chaînes de caractères (`"Hello World!"`)
- listes, arbres, *etc*

$\Longrightarrow$ largement utilisé dans les langages de programmation moderne : C, C++, Java, *etc*

Il faut manipuler la mémoire avec **précaution**



Manipulations de mémoire

C'est l'utilisation, en programmation, de

- tableaux, matrices
- chaînes de caractères (`"Hello World!"`)
- listes, arbres, *etc*

$\Longrightarrow$ largement utilisé dans les langages de programmation moderne : C, C++, Java, *etc*

Il faut manipuler la mémoire avec **précaution**
$\longrightarrow$ attention aux buffer overflows



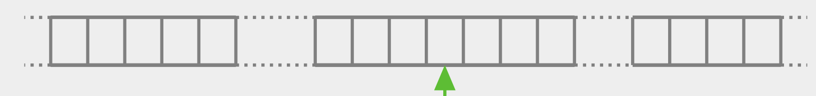
Manipulations de mémoire

C'est l'utilisation, en programmation, de

- tableaux, matrices
- chaînes de caractères (`"Hello World!"`)
- listes, arbres, *etc*

$\Longrightarrow$ largement utilisé dans les langages de programmation moderne : C, C++, Java, *etc*

Il faut manipuler la mémoire avec **précaution**
$\longrightarrow$ attention aux buffer overflows



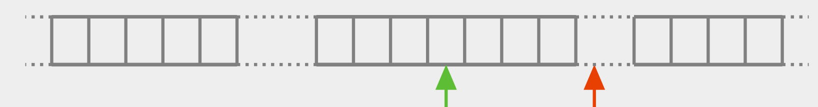
Manipulations de mémoire

C'est l'utilisation, en programmation, de

- tableaux, matrices
- chaînes de caractères (`"Hello World!"`)
- listes, arbres, *etc*

$\Longrightarrow$ largement utilisé dans les langages de programmation moderne : C, C++, Java, *etc*

Il faut manipuler la mémoire avec **précaution**
$\longrightarrow$ attention aux buffer overflows



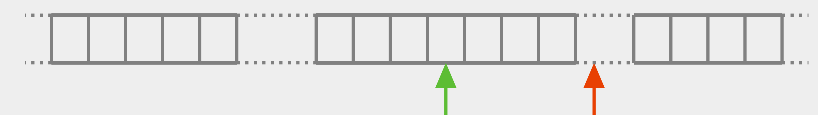
Manipulations de mémoire

C'est l'utilisation, en programmation, de

- tableaux, matrices
- chaînes de caractères (`"Hello World!"`)
- listes, arbres, *etc*

$\Longrightarrow$ largement utilisé dans les langages de programmation moderne : C, C++, Java, *etc*

Il faut manipuler la mémoire avec **précaution**
$\longrightarrow$ attention aux buffer overflows



Les buffer overflows peuvent provoquer :

- un plantage de la machine (SEGFAULT)
- des trous de sécurité

Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

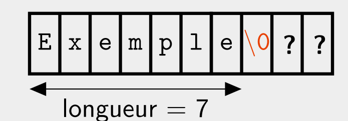
```
int i := 0;
for i = 0 to n-1 do
  dst[i] := src[i];
done;
```

Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

```
int i := 0;
for i = 0 to n-1 do
  dst[i] := src[i];
done;
```

Comment une chaîne de caractères est-elle codée en machine ?

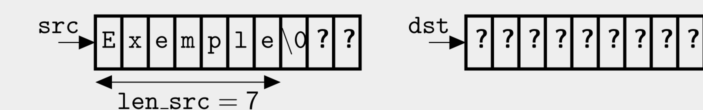


Manipulations de mémoire (2)

memcpy(dst,src,n) copie les n premiers caractères de src dans dst :

```
1: int i := 0;
2: for i = 0 to n-1 do
3:   dst[i] := src[i];
4: done;
```

- si $n > \mathtt{len_src}$,

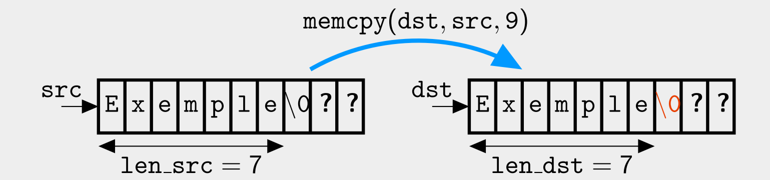


Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

```
int i := 0;
for i = 0 to n-1 do
  dst[i] := src[i];
done;
```

- si $\mathtt{n} > \mathtt{len_src}$,

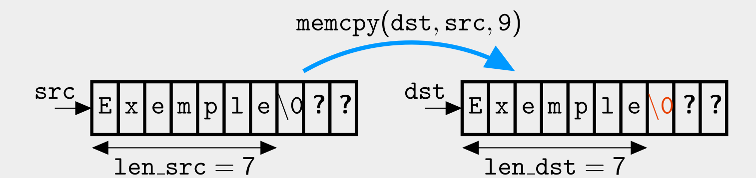


Manipulations de mémoire (2)

memcpy(dst,src,n) copie les n premiers caractères de src dans dst :

```
1: int i := 0;
2: for i = 0 to n-1 do
3:   dst[i] := src[i];
4: done;
```

- si $n > len_src$, $len_dst = len_src$

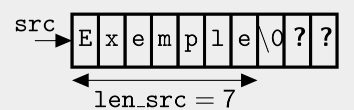


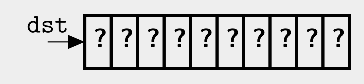
Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

```
int i := 0;
for i = 0 to n-1 do
  dst[i] := src[i];
done;
```

- si $\mathtt{n} > \mathtt{len_src}$, $\mathtt{len_dst} = \mathtt{len_src}$
- si $\mathtt{n} \leq \mathtt{len_src}$,



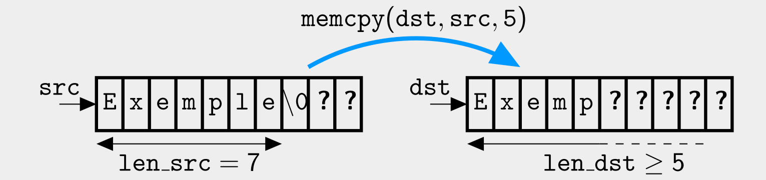


Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

```
1: int i := 0;
2: for i = 0 to n-1 do
3:   dst[i] := src[i];
4: done;
```

- si $\mathtt{n} > \mathtt{len_src}$, $\mathtt{len_dst} = \mathtt{len_src}$
- si $\mathtt{n} \leq \mathtt{len_src}$,

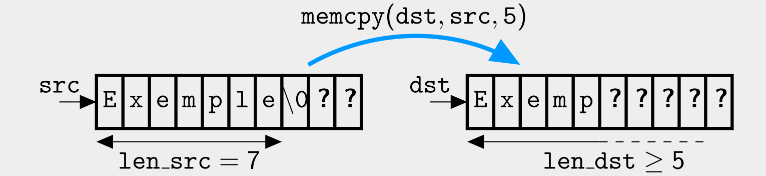


Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

```
1: int i := 0;
2: for i = 0 to n-1 do
3:   dst[i] := src[i];
4: done;
```

- si $\mathtt{n} > \mathtt{len_src}$, $\mathtt{len_dst} = \mathtt{len_src}$
- si $\mathtt{n} \leq \mathtt{len_src}$, $\mathtt{len_dst} \geq \mathtt{n}$



Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

```
int i := 0;
for i = 0 to n-1 do
  dst[i] := src[i];
done;
```

- si $\texttt{n} > \texttt{len_src}$, $\texttt{len_dst} = \texttt{len_src}$
- si $\texttt{n} \leq \texttt{len_src}$, $\texttt{len_dst} \geq \texttt{n}$

Manipulations de mémoire (2)

`memcpy(dst,src,n)` copie les `n` premiers caractères de `src` dans `dst` :

```
int i := 0;
for i = 0 to n-1 do
  dst[i] := src[i];
done;
```

- si $\mathtt{n} > \mathtt{len_src}$, $\mathtt{len_dst} = \mathtt{len_src}$
- si $\mathtt{n} \leq \mathtt{len_src}$, $\mathtt{len_dst} \geq \mathtt{n}$

$$\min(\mathtt{len_dst}, \mathtt{n}) = \min(\mathtt{len_src}, \mathtt{n})$$

Résumé des épisodes précédents

Nous avons vu que :

- les bugs sont une chose sérieuse

Résumé des épisodes précédents

Nous avons vu que :

- les bugs sont une chose sérieuse
- l'interprétation abstraite permet de faire de la vérification, en faisant des calculs sur des formes géométriques, qui approximent les comportements du programme

Résumé des épisodes précédents

Nous avons vu que :

- les bugs sont une chose sérieuse
- l'interprétation abstraite permet de faire de la vérification, en faisant des calculs sur des formes géométriques, qui approximent les comportements du programme
- les programmes utilisent (indirectement) des min et des max

Résumé des épisodes précédents

Nous avons vu que :

- les bugs sont une chose sérieuse
- l'interprétation abstraite permet de faire de la vérification, en faisant des calculs sur des formes géométriques, qui approximent les comportements du programme
- les programmes utilisent (indirectement) des min et des max

Problème :

- les propriétés avec des min et max sont de nature disjonctive

$$\max(x, y) \geq 2 \quad \text{équivaut à} \quad x \geq 2 \text{ ou } y \geq 2$$

Résumé des épisodes précédents

Nous avons vu que :

- les bugs sont une chose sérieuse
- l'interprétation abstraite permet de faire de la vérification, en faisant des calculs sur des formes géométriques, qui approximent les comportements du programme
- les programmes utilisent (indirectement) des min et des max

Problème :

- les propriétés avec des min et max sont de nature disjonctive

$$\max(x, y) \geq 2 \quad \text{équivaut à} \quad x \geq 2 \text{ ou } y \geq 2$$

- les techniques existantes en IA ne permettent pas bien d'approximer ces propriétés.

Nous allons voir comment la géométrie tropicale peut apporter des solutions à ce problème.

Plan de l'exposé

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$

$$1 \oplus 1 = \max(1, 1)$$
$$1 \otimes 1 =$$
$$2 \oplus 3 =$$
$$2 \otimes 3 =$$

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

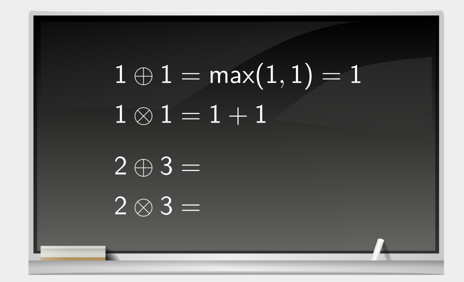
1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$

$$1 \oplus 1 = \max(1, 1) = 1$$
$$1 \otimes 1 =$$
$$2 \oplus 3 =$$
$$2 \otimes 3 =$$

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

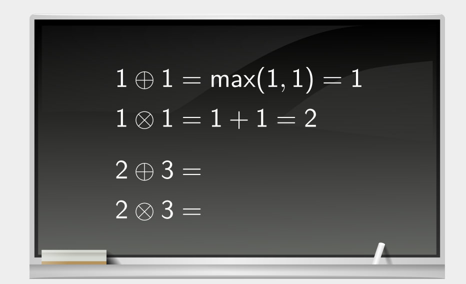
1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$



L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$



L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$

$$1 \oplus 1 = \max(1,1) = 1$$
$$1 \otimes 1 = 1 + 1 = 2$$
$$2 \oplus 3 = \max(2,3)$$
$$2 \otimes 3 =$$

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$

$$1 \oplus 1 = \max(1,1) = 1$$
$$1 \otimes 1 = 1 + 1 = 2$$
$$2 \oplus 3 = \max(2,3) = 3$$
$$2 \otimes 3 =$$

L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$

$$1 \oplus 1 = \max(1,1) = 1$$
$$1 \otimes 1 = 1 + 1 = 2$$
$$2 \oplus 3 = \max(2,3) = 3$$
$$2 \otimes 3 = 2 + 3$$

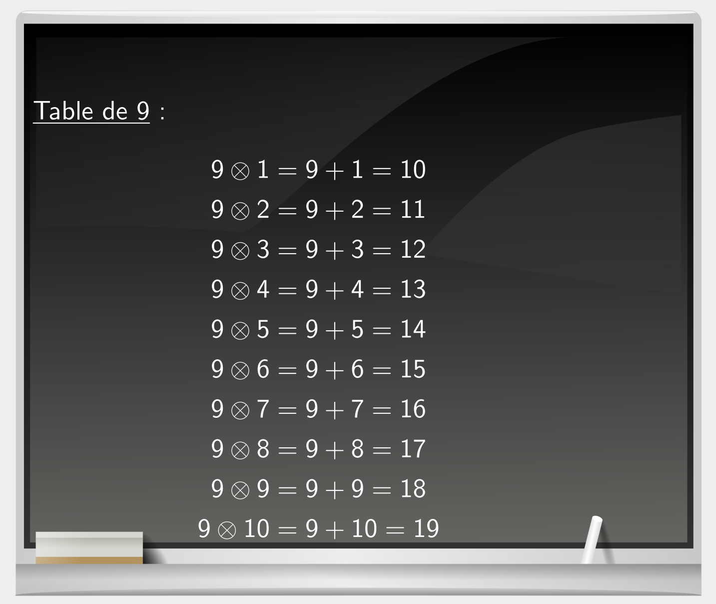
L'algèbre tropicale

Qu'est-ce que l'algèbre tropicale ?

1. Oubliez ce que vous avez appris à l'école primaire
2. l'addition $\oplus$ est maintenant l'opération max
3. la multiplication $\otimes$ correspond à l'opération $+$

$$1 \oplus 1 = \max(1,1) = 1$$
$$1 \otimes 1 = 1 + 1 = 2$$
$$2 \oplus 3 = \max(2,3) = 3$$
$$2 \otimes 3 = 2 + 3 = 5$$

Les tables de multiplication tropicale



L'algèbre tropicale a une jolie structure

- l'addition est commutative et associative

$$2 \oplus 3 = 3 \oplus 2 \qquad 2 \oplus (3 \oplus 4) = (2 \oplus 3) \oplus 4$$

L'algèbre tropicale a une jolie structure

- l'addition est commutative et associative

$$2 \oplus 3 = 3 \oplus 2 \qquad 2 \oplus (3 \oplus 4) = (2 \oplus 3) \oplus 4$$

- idem pour la multiplication :

$$3 \otimes 4 = 4 \otimes 3 \qquad 1 \otimes (3 \otimes 7) = (1 \otimes 3) \otimes 7$$

L'algèbre tropicale a une jolie structure

- l'addition est commutative et associative

$$2 \oplus 3 = 3 \oplus 2 \qquad 2 \oplus (3 \oplus 4) = (2 \oplus 3) \oplus 4$$

- idem pour la multiplication :

$$3 \otimes 4 = 4 \otimes 3 \qquad 1 \otimes (3 \otimes 7) = (1 \otimes 3) \otimes 7$$

- la multiplication est distributive par rapport à l'addition :

$$3 \otimes (2 \otimes 5) = (3 \otimes 2) \oplus (3 \otimes 5)$$

L'algèbre tropicale a une jolie structure

- l'addition est commutative et associative

$$2 \oplus 3 = 3 \oplus 2 \qquad 2 \oplus (3 \oplus 4) = (2 \oplus 3) \oplus 4$$

- idem pour la multiplication :

$$3 \otimes 4 = 4 \otimes 3 \qquad 1 \otimes (3 \otimes 7) = (1 \otimes 3) \otimes 7$$

- la multiplication est distributive par rapport à l'addition :

$$3 \otimes (2 \otimes 5) = (3 \otimes 2) \oplus (3 \otimes 5)$$

- il y a un "zero" et un "un" tropicaux : $\mathbb{0} := -\infty$, $\mathbb{1} := 0$:

$$3 \oplus \mathbb{0} = 3 \qquad 4 \otimes \mathbb{1} = \mathbb{1}$$

L'algèbre tropicale a une jolie structure

- l'addition est commutative et associative

$$2 \oplus 3 = 3 \oplus 2 \qquad 2 \oplus (3 \oplus 4) = (2 \oplus 3) \oplus 4$$

- idem pour la multiplication :

$$3 \otimes 4 = 4 \otimes 3 \qquad 1 \otimes (3 \otimes 7) = (1 \otimes 3) \otimes 7$$

- la multiplication est distributive par rapport à l'addition :

$$3 \otimes (2 \otimes 5) = (3 \otimes 2) \oplus (3 \otimes 5)$$

- il y a un "zero" et un "un" tropicaux : $\mathbb{0} := -\infty$, $\mathbb{1} := 0$:

$$3 \oplus \mathbb{0} = 3 \qquad 4 \otimes \mathbb{1} = \mathbb{1}$$

- il y a une division $x \oslash y := x - y$:

$$3 \oslash 3 = 3 - 3 = 0 = \mathbb{1}$$

L'algèbre tropicale a une jolie structure

- l'addition est commutative et associative

$$2 \oplus 3 = 3 \oplus 2 \qquad 2 \oplus (3 \oplus 4) = (2 \oplus 3) \oplus 4$$

- idem pour la multiplication :

$$3 \otimes 4 = 4 \otimes 3 \qquad 1 \otimes (3 \otimes 7) = (1 \otimes 3) \otimes 7$$

- la multiplication est distributive par rapport à l'addition :

$$3 \otimes (2 \otimes 5) = (3 \otimes 2) \oplus (3 \otimes 5)$$

- il y a un "zero" et un "un" tropicaux : $\mathbb{0} := -\infty$, $\mathbb{1} := 0$:

$$3 \oplus \mathbb{0} = 3 \qquad 4 \otimes \mathbb{1} = \mathbb{1}$$

- il y a une division $x \oslash y := x - y$:

$$3 \oslash 3 = 3 - 3 = 0 = \mathbb{1}$$

- **MAIS** il n'y a pas de soustraction :

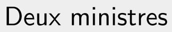
$$1 \oplus x = \max(1, x) = \mathbb{0} = -\infty$$

  Que pourrait valoir x ?

L'algèbre des ordres de grandeur

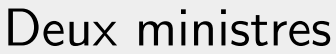
Deux ministres   sont chez le président.

L'algèbre des ordres de grandeur


Deux ministres   sont chez le président.

Le président : *J'ai trois super idées pour combler le déficit :*

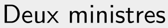
L'algèbre des ordres de grandeur


Deux ministres   sont chez le président.

Le président : *J'ai trois super idées pour combler le déficit :*

1. *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*

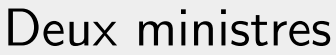
L'algèbre des ordres de grandeur


Deux ministres   sont chez le président.

Le président : *J'ai trois super idées pour combler le déficit :*

1. *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
2. *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*

L'algèbre des ordres de grandeur

 Deux ministres   sont chez le président.

Le président : *J'ai trois super idées pour combler le déficit :*

1. *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
2. *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
3. *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*

L'algèbre des ordres de grandeur

Deux ministres   sont chez le président.

Le président : *J'ai trois super idées pour combler le déficit :*

1. *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
2. *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
3. *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*

Combien en gros ça va rapporter ?

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$12 \times 14\,998\,462 =$$
$$2.54 \times 15\,214\,261 =$$
$$151.23 \times 190\,217 =$$
$$\text{Total} =$$

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$
\begin{aligned}
12 \times 14\,998\,462 &= 179\,981\,544 \\
2.54 \times 15\,214\,261 &= \\
151.23 \times 190\,217 &= \\
\text{Total} &=
\end{aligned}
$$

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$12 \times 14\,998\,462 = 179\,981\,544 \qquad \approx 10 \times 10\,000\,000$$

$$2.54 \times 15\,214\,261 =$$

$$151.23 \times 190\,217 =$$

$$\text{Total} =$$

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$12 \times 14\,998\,462 = 179\,981\,544 \qquad \approx 10 \times 10\,000\,000 = 100\,000\,000$$

$$2.54 \times 15\,214\,261 =$$

$$151.23 \times 190\,217 =$$

$$\text{Total} =$$

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$12 \times 14\,998\,462 = 179\,981\,544 \quad \approx 10 \times 10\,000\,000 = 100\,000\,000$$

$$2.54 \times 15\,214\,261 = \quad \approx 1 \times 10\,000\,000 = 10\,000\,000$$

$$151.23 \times 190\,217 =$$

$$\text{Total} =$$

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$12 \times 14\,998\,462 = 179\,981\,544 \quad \approx 10 \times 10\,000\,000 = 100\,000\,000$$

$$2.54 \times 15\,214\,261 = \quad \approx 1 \times 10\,000\,000 = 10\,000\,000$$

$$151.23 \times 190\,217 = \quad \approx 100 \times 100\,000 = 10\,000\,000$$

$$\text{Total} =$$

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$12 \times 14\,998\,462 = 179\,981\,544 \qquad \approx 10 \times 10\,000\,000 = 100\,000\,000$$

$$2.54 \times 15\,214\,261 = \qquad \approx 1 \times 10\,000\,000 = 10\,000\,000$$

$$151.23 \times 190\,217 = \qquad \approx 100 \times 100\,000 = 10\,000\,000$$

$$\text{Total} = \qquad \text{Total} \approx 100\,000\,000$$

Temps de calcul : 2s

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$
\begin{aligned}
12 \times 14\,998\,462 &= 179\,981\,544 \\
2.54 \times 15\,214\,261 &= 38\,644\,222.94 \\
151.23 \times 190\,217 &= \\
\text{Total} &=
\end{aligned}
$$

$$
\begin{aligned}
\approx 10 \times 10\,000\,000 &= 100\,000\,000 \\
\approx 1 \times 10\,000\,000 &= 10\,000\,000 \\
\approx 100 \times 100\,000 &= 10\,000\,000 \\
\text{Total} &\approx 100\,000\,000
\end{aligned}
$$

Temps de calcul : 2s

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$12 \times 14\,998\,462 = 179\,981\,544$$
$$2.54 \times 15\,214\,261 = 38\,644\,222.94$$
$$151.23 \times 190\,217 = 28\,766\,516.91$$
$$\text{Total} =$$

$$\approx 10 \times 10\,000\,000 = 100\,000\,000$$
$$\approx 1 \times 10\,000\,000 = 10\,000\,000$$
$$\approx 100 \times 100\,000 = 10\,000\,000$$
$$\text{Total} \approx 100\,000\,000$$

Temps de calcul : 2s

L'algèbre des ordres de grandeur

- *contribution forfaitaire de 12 euros pour chaque élève ou étudiant, pour le désendettement futur de la France*
- *taxe de 2.54 euros sur chaque entrée à Disneyland Paris*
- *taxation des emplacements de camping 4 étoiles, 151.23 euros par emplacement*





$$\begin{aligned} 12 \times 14\,998\,462 &= 179\,981\,544 \\ 2.54 \times 15\,214\,261 &= 38\,644\,222.94 \\ 151.23 \times 190\,217 &= 28\,766\,516.91 \\ \text{Total} &= 247\,392\,283.85 \end{aligned}$$

$$\begin{aligned} \approx 10 \times 10\,000\,000 &= 100\,000\,000 \\ \approx 1 \times 10\,000\,000 &= 10\,000\,000 \\ \approx 100 \times 100\,000 &= 10\,000\,000 \\ \text{Total} &\approx 100\,000\,000 \end{aligned}$$

Temps de calcul : 17s ! ! !

Temps de calcul : 2s

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

$$\begin{aligned} 10 \times 10\,000\,000 &= 100\,000\,000 \\ 1 \times 10\,000\,000 &= 10\,000\,000 \\ 100 \times 100\,000 &= 10\,000\,000 \\ \text{Total} &\approx 100\,000\,000 \end{aligned}$$

L'algèbre des ordres de grandeur (2)


fait des calculs en algèbre tropicale :

$$10^{1} \times 10^{7} = 10^{1+7}$$
$$1 \times 10\,000\,000 = 10\,000\,000$$
$$100 \times 100\,000 = 10\,000\,000$$
$$\text{Total} \approx 100\,000\,000$$

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

$$\begin{aligned} 10^{1} \times 10^{7} &= 10^{8} \\ 1 \times 10\,000\,000 &= 10\,000\,000 \\ 100 \times 100\,000 &= 10\,000\,000 \\ \text{Total} &\approx 100\,000\,000 \end{aligned}$$

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

$$10^{1} \times 10^{7} = 10^{8}$$

$$10^{0} \times 10^{7} = 10^{7}$$

$$100 \times 100\,000 = 10\,000\,000$$

$$\text{Total} \approx 100\,000\,000$$

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

$$10^{1} \times 10^{7} = 10^{8}$$
$$10^{0} \times 10^{7} = 10^{7}$$
$$10^{2} \times 10^{5} = 10^{7}$$
$$\text{Total} \approx 100\,000\,000$$

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

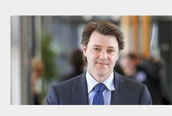
$$10^{1} \times 10^{7} = 10^{8}$$
$$10^{0} \times 10^{7} = 10^{7}$$
$$10^{2} \times 10^{5} = 10^{7}$$
$$\text{Total} \approx 10^{\max(8,7,7)}$$

L'algèbre des ordres de grandeur (2)

 fait des calculs en algèbre tropicale :

$$10^{1} \times 10^{7} = 10^{8}$$
$$10^{0} \times 10^{7} = 10^{7}$$
$$10^{2} \times 10^{5} = 10^{7}$$
$$\text{Total} \approx 10^{8}$$

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

$$10^{1} \times 10^{7} = 10^{8}$$
$$10^{0} \times 10^{7} = 10^{7}$$
$$10^{2} \times 10^{5} = 10^{7}$$
$$\text{Total} \approx 10^{8}$$

Autrement dit :

$$1 \otimes 7 = 8$$
$$0 \otimes 7 = 7$$
$$2 \otimes 5 = 7$$
$$\text{Total} = 8 \oplus 7 \oplus 7 = 8$$

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

$$10^{1} \times 10^{7} = 10^{8}$$
$$10^{0} \times 10^{7} = 10^{7}$$
$$10^{2} \times 10^{5} = 10^{7}$$
$$\text{Total} \approx 10^{8}$$

Le calcul est approximatif avec les puissances de 10, mais moins avec les puissances de β, pour β choisi grand :

$$\max(x, y) \leq \log_\beta(\beta^x + \beta^y) \leq \max(x, y) + \log_\beta 2$$

L'algèbre des ordres de grandeur (2)



fait des calculs en algèbre tropicale :

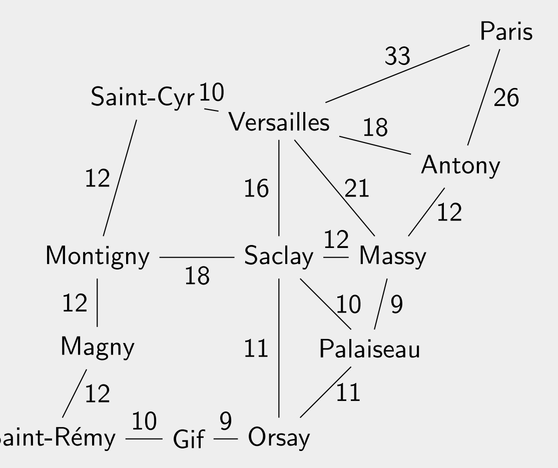
$$10^{1} \times 10^{7} = 10^{8}$$
$$10^{0} \times 10^{7} = 10^{7}$$
$$10^{2} \times 10^{5} = 10^{7}$$
$$\text{Total} \approx 10^{8}$$

Le calcul est approximatif avec les puissances de 10, mais moins avec les puissances de β, pour β choisi grand :

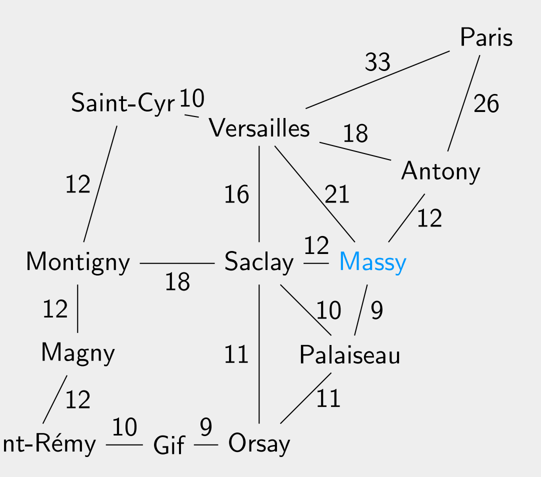
$$\max(x, y) \leq \log_\beta(\beta^x + \beta^y) \leq \max(x, y) + \log_\beta 2$$

$$\log_\beta(\beta^x \times \beta^y) = x + y$$

L'algèbre des plus courts chemins



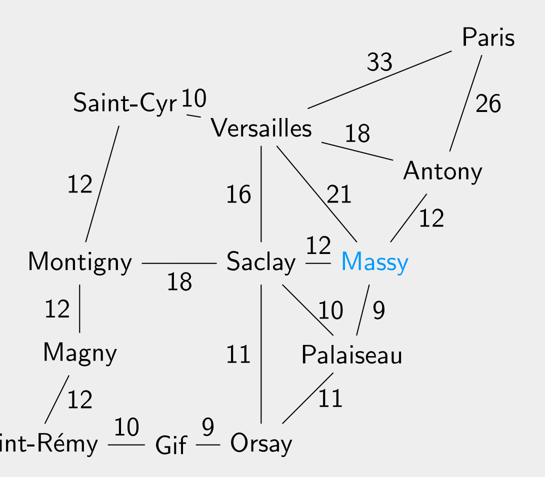
L'algèbre des plus courts chemins



Le plus court chemin entre de Massy à Magny est donné par :

$$\min(d_{\text{Saclay-Magny}} + 12, d_{\text{Pal.-Magny}} + 9, d_{\text{Ver.-Magny}} + 21, d_{\text{Antony-Magny}} + 12)$$

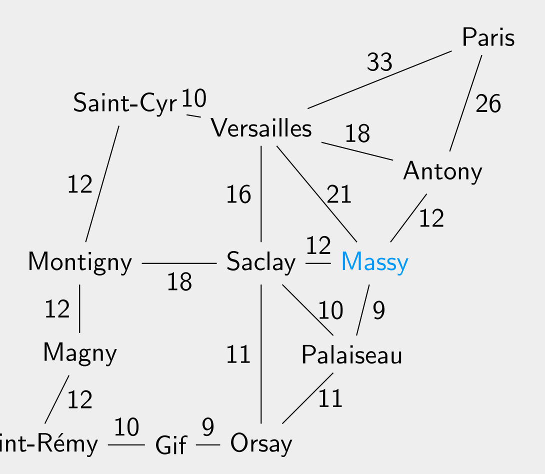
L'algèbre des plus courts chemins



L'opposé du plus court chemin est donné par :

$\max(-d_{\text{Saclay-Magny}} - 12, -d_{\text{Pal.-Magny}} - 9, -d_{\text{Ver.-Magny}} - 21, -d_{\text{Antony-Magny}} - 12)$

L'algèbre des plus courts chemins



L'opposé du plus court chemin est donné par :

$$(-12) \otimes (-d_{\text{Saclay-Magny}}) \ \oplus \ (-9) \otimes (-d_{\text{Pal.-Magny}}) \ \oplus \ (-21) \otimes (-d_{\text{Ver.-Magny}}) \ \oplus \ (-12) \otimes (-d_{\text{Antony-Magny}})$$

L'algèbre des plus courts chemins (2)

Plus généralement, si A est la matrice composée de l'opposé des distances :

$$A_{ij} = \text{opposé de la distance entre les villes } i \text{ et } j$$

L'algèbre des plus courts chemins (2)

Plus généralement, si A est la matrice composée de l'opposé des distances :

$$A_{ij} = \text{opposé de la distance entre les villes } i \text{ et } j$$

alors l'opposé des plus courts chemins est donné par la matrice

$$A^* = A^0 \oplus A^1 \oplus A^2 \oplus \dots$$

où $A^n = \underbrace{A \otimes A \otimes \cdots \otimes A}_{n \text{ fois}}$ représente les plus courts chemins de longueurs au plus n.

L'algèbre des plus courts chemins (2)

Plus généralement, si A est la matrice composée de l'opposé des distances :

$$A_{ij} = \text{opposé de la distance entre les villes } i \text{ et } j$$

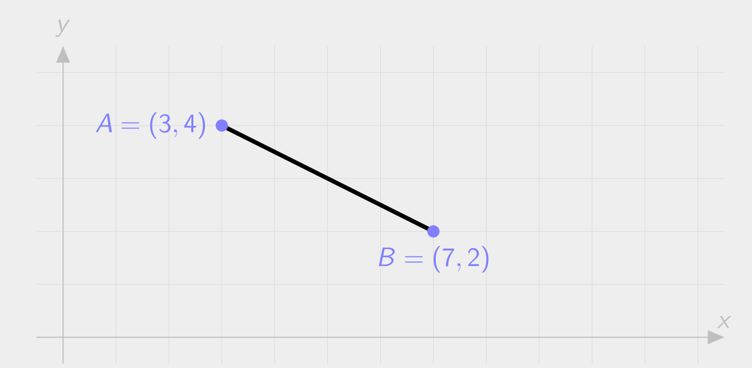
alors l'opposé des plus courts chemins est donné par la matrice

$$A^* = A^0 \oplus A^1 \oplus A^2 \oplus \cdots \oplus A^N$$

où $A^n = \underbrace{A \otimes A \otimes \cdots \otimes A}_{n \text{ fois}}$ représente les plus courts chemins de longueurs au plus n.

Plan de l'exposé

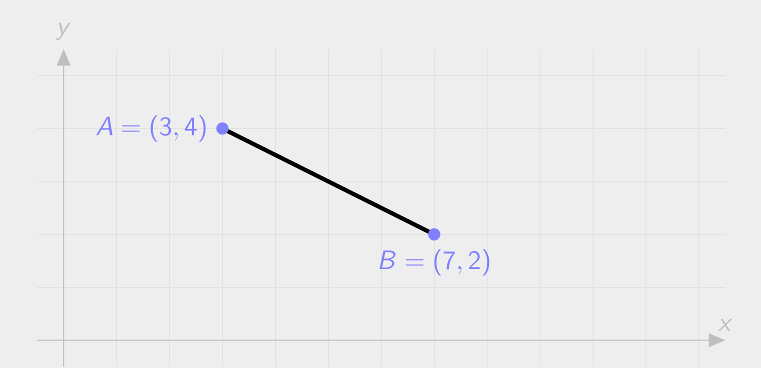
Le cas du segment de droite



Le cas du segment de droite

Le segment entre A et B est formé des barycentres $\lambda \times A + \mu \times B$, où λ, μ sont positifs, et $\lambda + \mu = 1$. Ici :

$$\lambda \times (3,4) + \mu \times (7,2) = (3\lambda + 7\mu, 4\lambda + 2\mu)$$



Le cas du segment de droite

Le segment entre A et B est formé des barycentres $\lambda \times A + \mu \times B$, où λ, μ sont positifs, et $\lambda + \mu = 1$. Ici :

$$\lambda \times (3,4) + \mu \times (7,2) = (3\lambda + 7\mu, 4\lambda + 2\mu)$$

Le cas du segment de droite (2)

Le segment *tropical* entre A et B est formé des barycentres $\lambda \otimes A \oplus \mu \otimes B$, où $\lambda \oplus \mu = \mathbb{1}$. Ici :

$$\lambda \otimes (3,4) \oplus \mu \otimes (7,2) = (\ \max(3+\lambda, 7+\mu)\ ,\ \max(4+\lambda, 2+\mu)\ )$$

Le cas du segment de droite (2)

Le segment *tropical* entre A et B est formé des barycentres $\lambda \otimes A \oplus \mu \otimes B$, où $\lambda \oplus \mu = \mathbb{1}$. Ici :

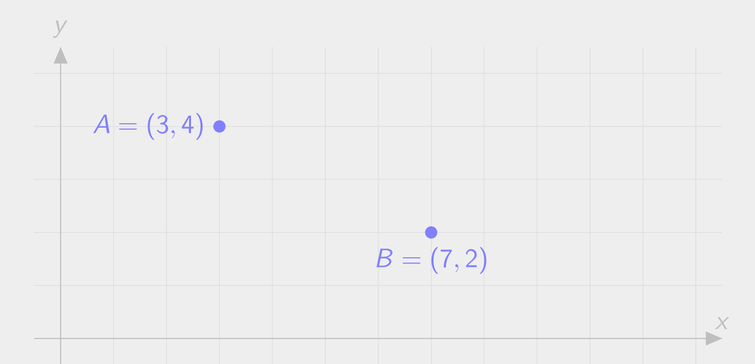
$$\lambda \otimes (3,4) \oplus \mu \otimes (7,2) = (\ \max(3+\lambda, 7+\mu)\ ,\ \max(4+\lambda, 2+\mu)\ )$$

- la condition de positivité $\lambda, \mu \geq \mathbb{0} = -\infty$ est toujours satisfaite,
- $\lambda \oplus \mu = \mathbb{1}$ revient à dire que $\max(\lambda, \mu) = 0$, soit $\lambda, \mu \leq 0$, et l'un des deux doit être nul !

Le cas du segment de droite (2)

Le segment *tropical* entre A et B est formé des barycentres $\lambda \otimes A \oplus \mu \otimes B$, où $\lambda \oplus \mu = \mathbb{1}$. Ici :

$$\lambda \otimes (3,4) \oplus \mu \otimes (7,2) = (\ \max(3+\lambda, 7+\mu)\ ,\ \max(4+\lambda, 2+\mu)\ )$$

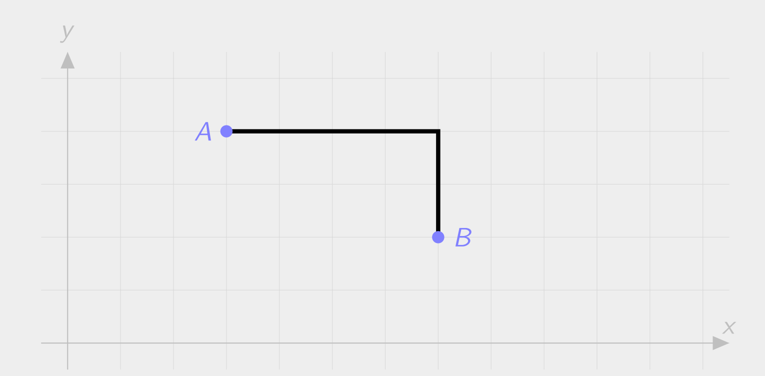


Le cas du segment de droite (2)

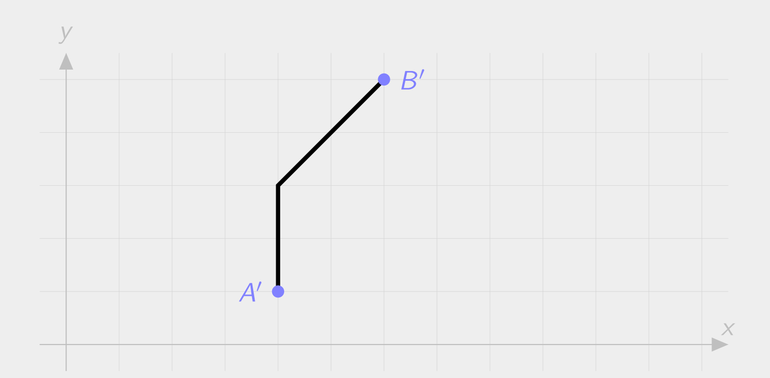
Le segment *tropical* entre A et B est formé des barycentres $\lambda \otimes A \oplus \mu \otimes B$, où $\lambda \oplus \mu = \mathbb{1}$. Ici :

$$\lambda \otimes (3,4) \oplus \mu \otimes (7,2) = (\ \max(3+\lambda, 7+\mu)\ ,\ \max(4+\lambda, 2+\mu)\ )$$

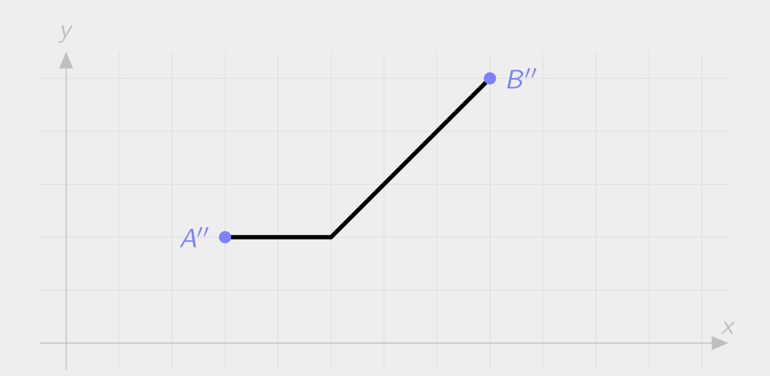
Le cas du segment de droite (3)



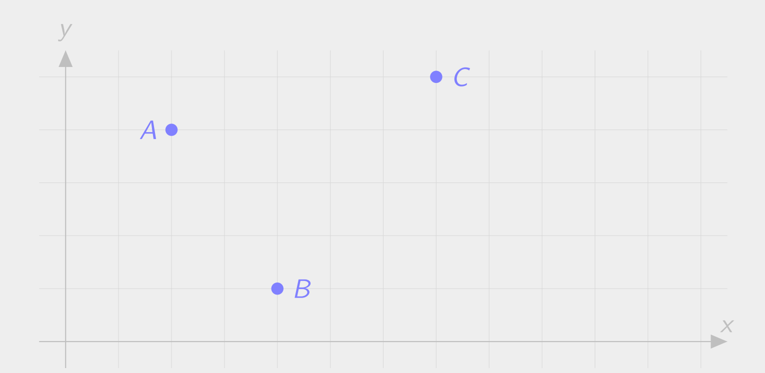
Le cas du segment de droite (3)



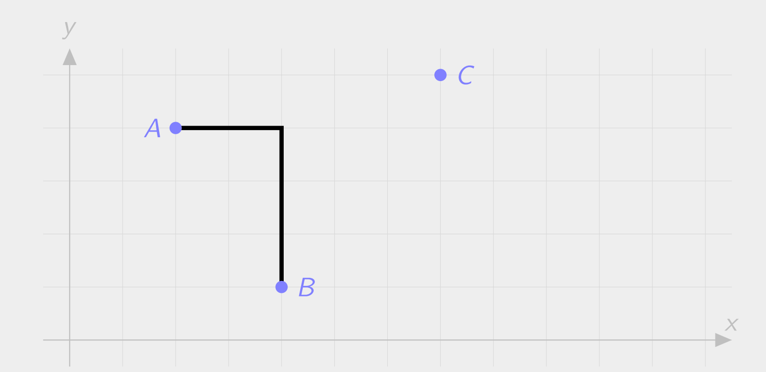
Le cas du segment de droite (3)



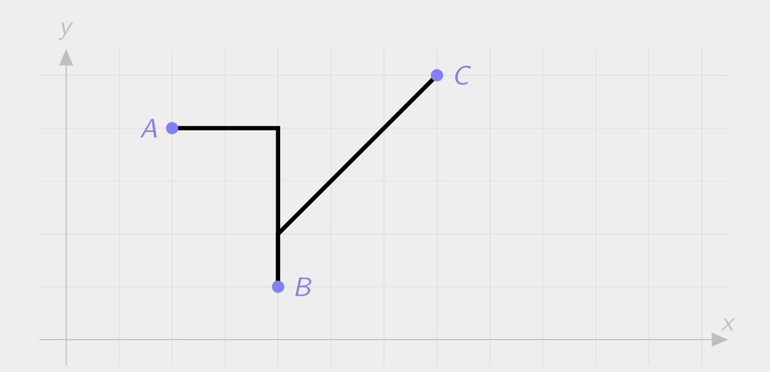
Les triangles tropicaux



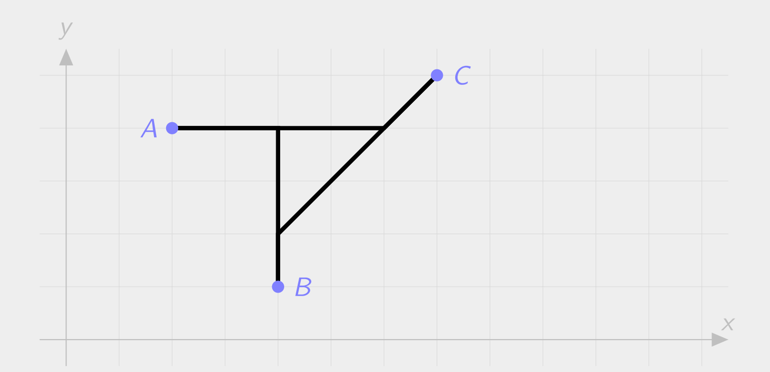
Les triangles tropicaux



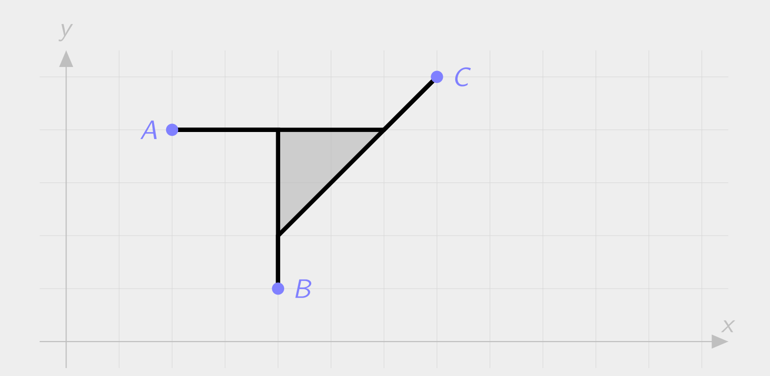
Les triangles tropicaux



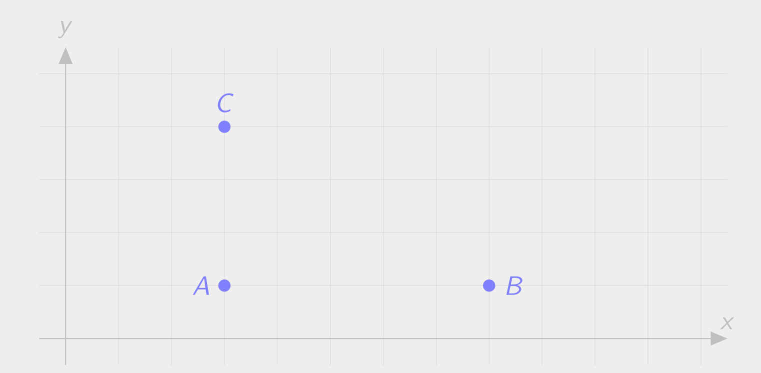
Les triangles tropicaux



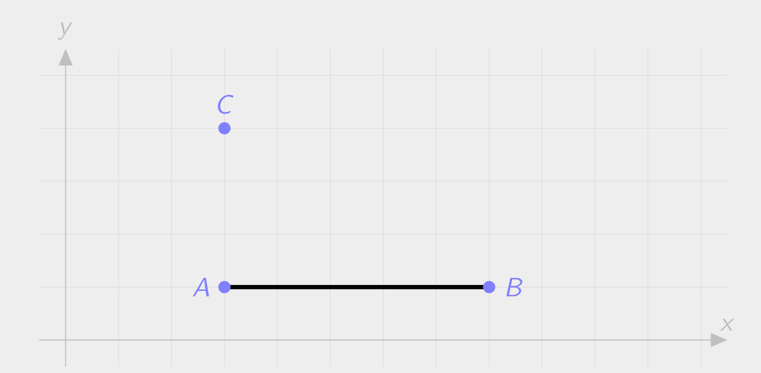
Les triangles tropicaux



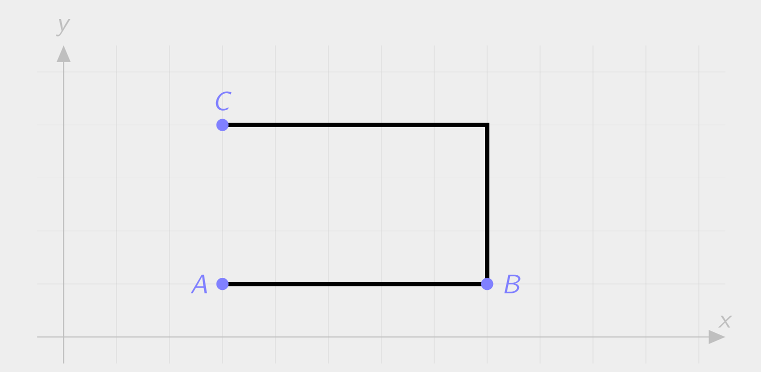
Les triangles tropicaux (2)



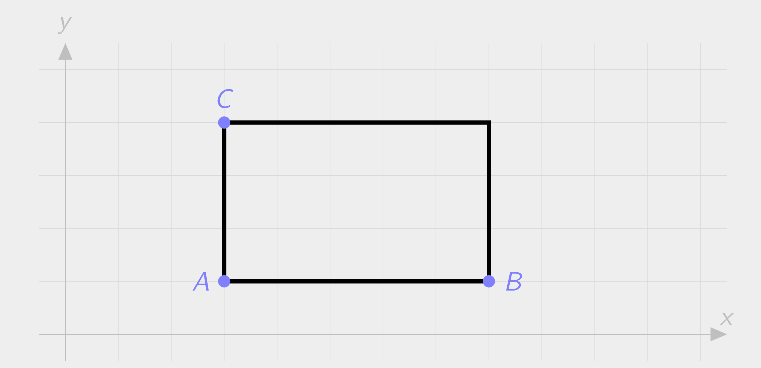
Les triangles tropicaux (2)



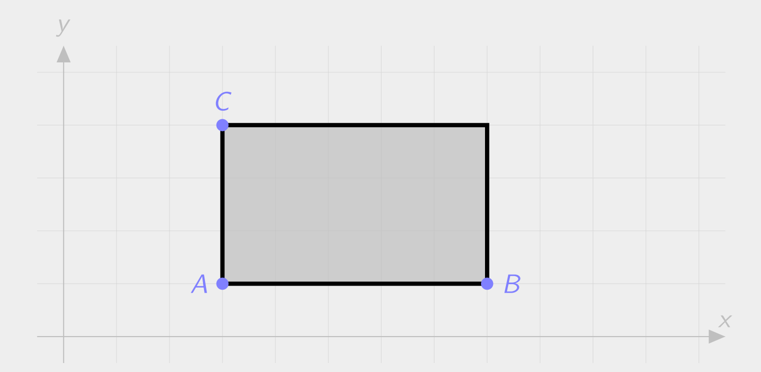
Les triangles tropicaux (2)



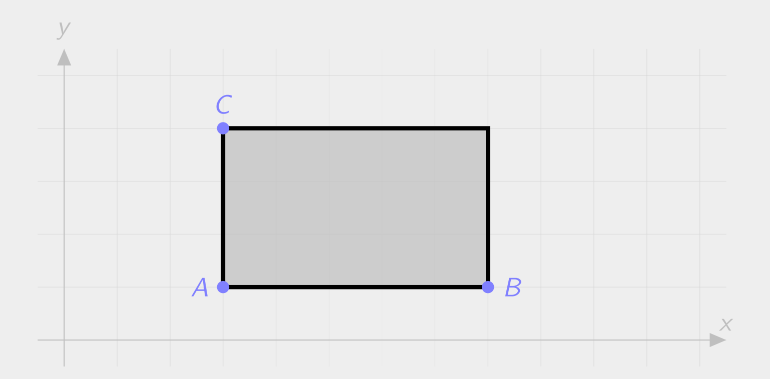
Les triangles tropicaux (2)



Les triangles tropicaux (2)



Les triangles tropicaux (2)



Plus généralement, les hyperrectangles en dimension n sont des simplexes tropicaux, *i.e.* des enveloppes tropicalement convexes de $n+1$ points.

Le cas des droites

Une droite classique :

$$aX + bY + c = 0$$

Le cas des droites

Une droite classique :

$$X + Y + 1 = 0$$

Le cas des droites

Une droite classique :

$$X + Y + 1 = 0$$

La droite tropicale est donnée par l'égalité sur les ordres de grandeurs

$$X = \Theta(\beta^{x}) \quad Y = \Theta(\beta^{y}) \quad \text{pour } \beta \text{ grand}$$

Le cas des droites

Une droite classique :

$$X + Y + 1 = 0$$

La droite tropicale est donnée par l'égalité sur les ordres de grandeurs

$$X = \Theta(\beta^{x}) \quad Y = \Theta(\beta^{y}) \quad \text{pour } \beta \text{ grand}$$

Le cas des droites

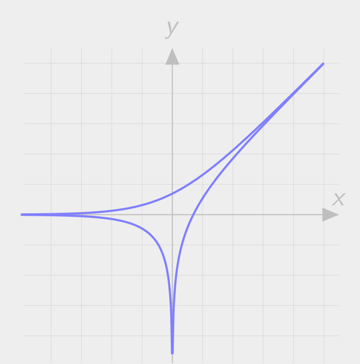
Une droite classique :

$$X + Y + 1 = 0$$

La droite tropicale est donnée par l'égalité sur les ordres de grandeurs

$$X = \Theta(\beta^{x}) \quad Y = \Theta(\beta^{y}) \quad \text{pour } \beta \text{ grand}$$

On regarde la droite $X + Y + 1 = 0$ avec des "lunettes logarithmiques", *i.e.* son image par l'application $(X, Y) \mapsto (\log_\beta |X|, \log_\beta |Y|)$



Le cas des droites

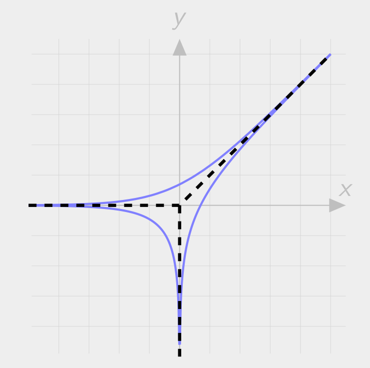
Une droite classique :

$$X + Y + 1 = 0$$

La droite tropicale est donnée par l'égalité sur les ordres de grandeurs

$$X = \Theta(\beta^{x}) \quad Y = \Theta(\beta^{y}) \quad \text{pour } \beta \text{ grand}$$

On regarde la droite $X + Y + 1 = 0$ avec des "lunettes logarithmiques", *i.e.* son image par l'application $(X, Y) \mapsto (\log_\beta |X|, \log_\beta |Y|)$



La droite tropicale est le "squelette" de l'amibe ($\beta \to +\infty$).

Le cas des droites

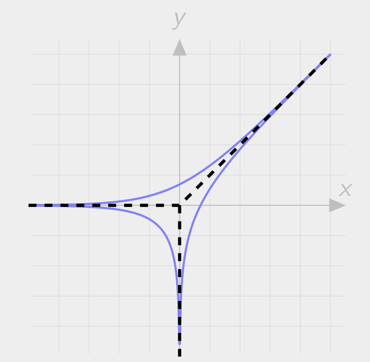
Une droite classique :

$$X + Y + 1 = 0$$

La droite tropicale est donnée par l'égalité sur les ordres de grandeurs

$$X = \Theta(\beta^{x}) \quad Y = \Theta(\beta^{y}) \quad \text{pour } \beta \text{ grand}$$

On regarde la droite $X + Y + 1 = 0$ avec des "lunettes logarithmiques", *i.e.* son image par l'application $(X, Y) \mapsto (\log_\beta |X|, \log_\beta |Y|)$



La droite tropicale est le "squelette" de l'amibe ($\beta \to +\infty$).

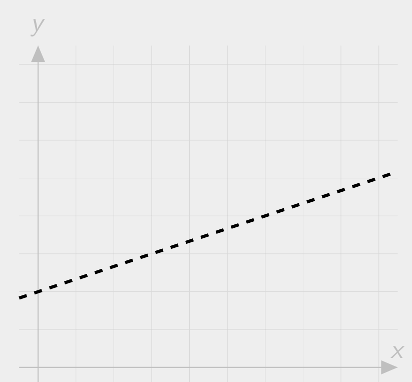
Elle est donnée par les points (x, y) tels que le maximum

$$\max(x, y, 0)$$

est atteint deux fois.

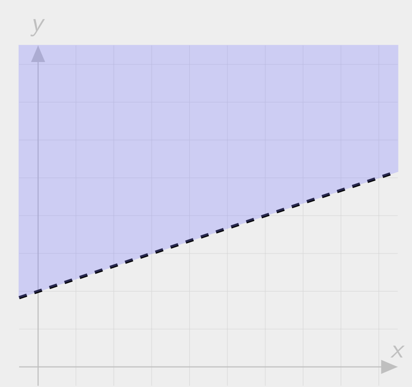
Les demi-espaces

Ils sont définis comme les ensembles situés “d’un côté” d’une droite (en dimension 2).



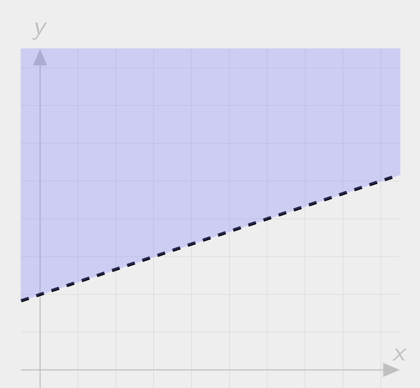
Les demi-espaces

Ils sont définis comme les ensembles situés “d’un côté” d’une droite (en dimension 2).



Les demi-espaces

Ils sont définis comme les ensembles situés “d'un côté” d'une droite (en dimension 2).

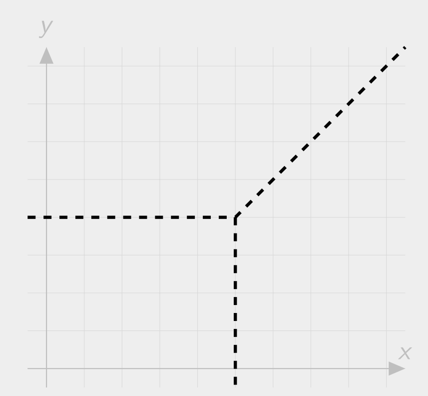


Un demi-espace est défini par une inégalité *affine*, ici :

$$y \geq (1/3) \times x + 2$$

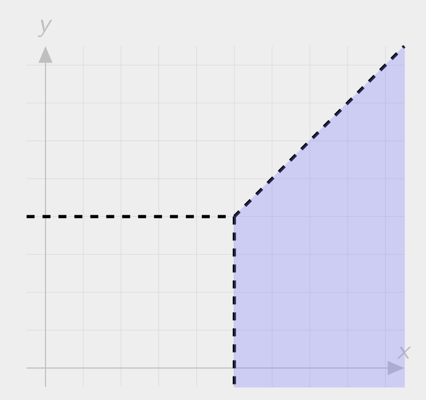
Les demi-espaces (2)

Une droite tropicale sépare le plan en 3 secteurs.



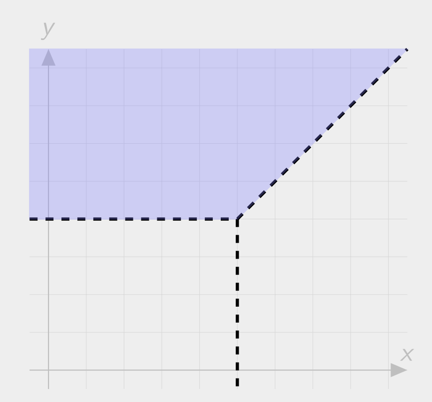
Les demi-espaces (2)

Une droite tropicale sépare le plan en 3 secteurs.



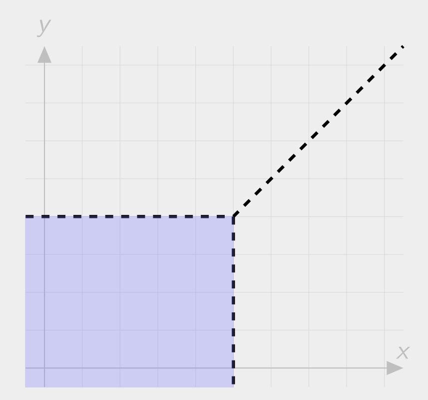
Les demi-espaces (2)

Une droite tropicale sépare le plan en 3 secteurs.



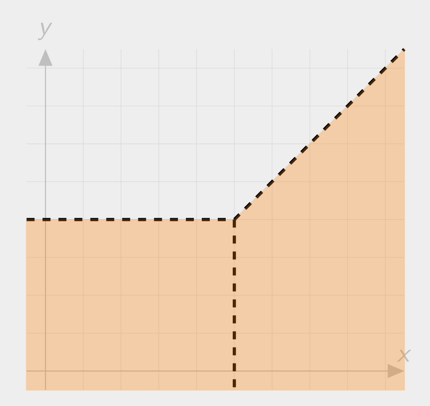
Les demi-espaces (2)

Une droite tropicale sépare le plan en 3 secteurs.



Les demi-espaces (2)

Une droite tropicale sépare le plan en 3 secteurs.

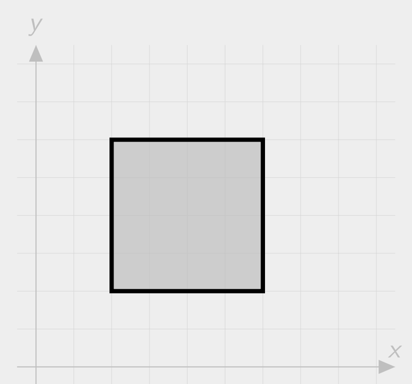


- les demi-espaces sont obtenus en sélectionnant 1 ou 2 secteurs
- ils correspondent aux solutions d'inégalités *tropicalement affines*

$$(-4) \otimes y \leq (-5) \otimes x \oplus 0$$

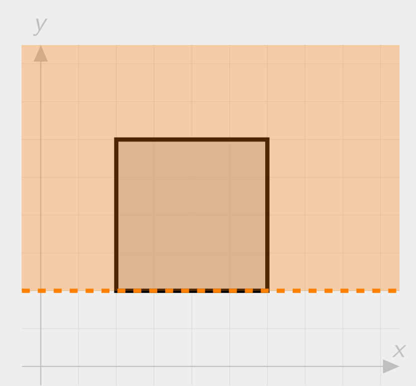
Les polyèdres (convexes)

On les utilisent depuis la maternelle : les triangles, les carrés, les rectangles, les losanges, les trapèzes, certains pentagones, hexagones, *etc*.



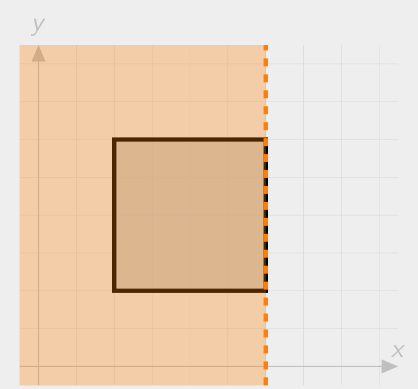
Les polyèdres (convexes)

On les utilisent depuis la maternelle : les triangles, les carrés, les rectangles, les losanges, les trapèzes, certains pentagones, hexagones, *etc*.



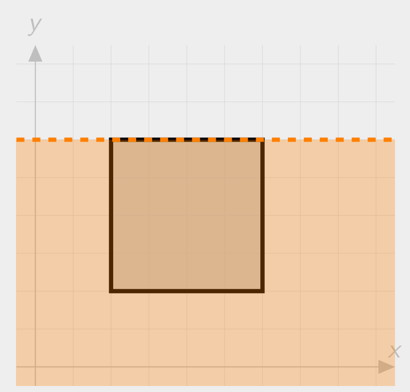
Les polyèdres (convexes)

On les utilisent depuis la maternelle : les triangles, les carrés, les rectangles, les losanges, les trapèzes, certains pentagones, hexagones, *etc*.



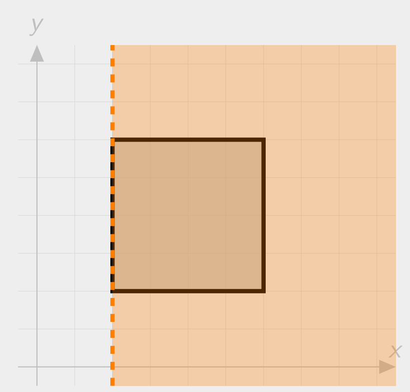
Les polyèdres (convexes)

On les utilisent depuis la maternelle : les triangles, les carrés, les rectangles, les losanges, les trapèzes, certains pentagones, hexagones, *etc*.



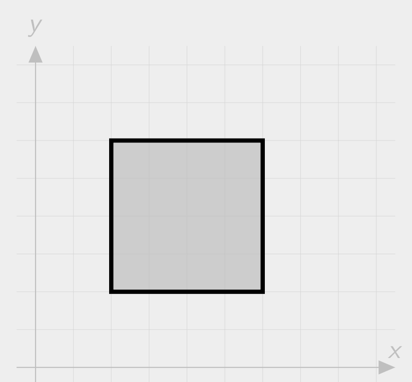
Les polyèdres (convexes)

On les utilisent depuis la maternelle : les triangles, les carrés, les rectangles, les losanges, les trapèzes, certains pentagones, hexagones, *etc*.



Les polyèdres (convexes)

On les utilisent depuis la maternelle : les triangles, les carrés, les rectangles, les losanges, les trapèzes, certains pentagones, hexagones, *etc*.



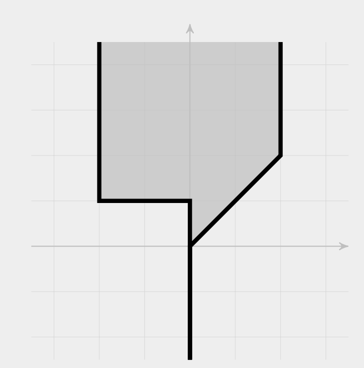
Ils sont donnés par les solutions de plusieurs inégalités (affines) :

$$2 \leq x \leq 6 \qquad 2 \leq y \leq 6$$

Les polyèdres tropicaux

Ils sont donnés par les solutions de plusieurs inégalités (affines) tropicales :

$$x \leq y \oplus 0$$
$$0 \leq 2 \otimes x$$
$$x \leq 2$$
$$0 \leq x \oplus (-1) \otimes y$$



Les polyèdres tropicaux

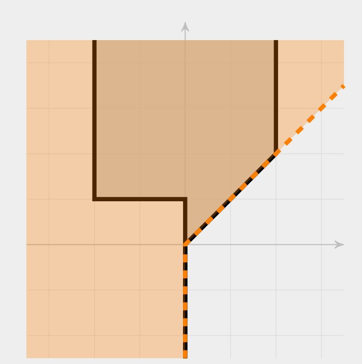
Ils sont donnés par les solutions de plusieurs inégalités (affines) tropicales :

$x \leq y \oplus 0$

$0 \leq 2 \otimes x$

$x \leq 2$

$0 \leq x \oplus (-1) \otimes y$



Les polyèdres tropicaux

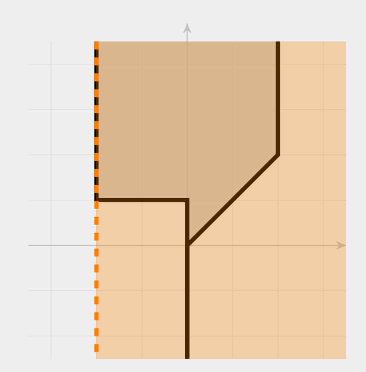
Ils sont donnés par les solutions de plusieurs inégalités (affines) tropicales :

$$x \leq y \oplus 0$$

$$0 \leq 2 \otimes x$$

$$x \leq 2$$

$$0 \leq x \oplus (-1) \otimes y$$



Les polyèdres tropicaux

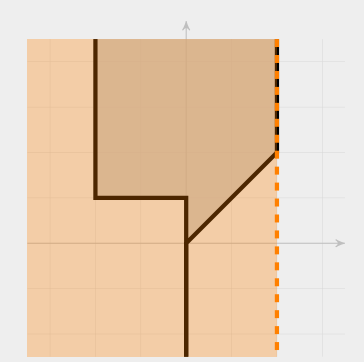
Ils sont donnés par les solutions de plusieurs inégalités (affines) tropicales :

$x \leq y \oplus 0$

$0 \leq 2 \otimes x$

$x \leq 2$

$0 \leq x \oplus (-1) \otimes y$



Les polyèdres tropicaux

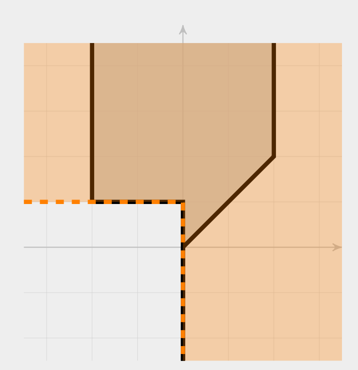
Ils sont donnés par les solutions de plusieurs inégalités (affines) tropicales :

$x \leq y \oplus 0$

$0 \leq 2 \otimes x$

$x \leq 2$

$0 \leq x \oplus (-1) \otimes y$



Les polyèdres tropicaux (2)

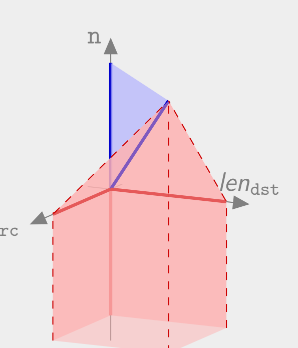
Ils permettent de représenter les propriétés avec min et max qu'on souhaite vérifier sur les programmes.

Exemple :

$$\min(\texttt{len_dst}, \texttt{n}) = \min(\texttt{len_src}, \texttt{n})$$

revient à :

$$\max(-\texttt{len_dst}, -\texttt{n}) \leq \max(-\texttt{len_src}, -\texttt{n})$$
$$\max(-\texttt{len_src}, -\texttt{n}) \leq \max(-\texttt{len_dst}, -\texttt{n})$$



Les polyèdres tropicaux (2)

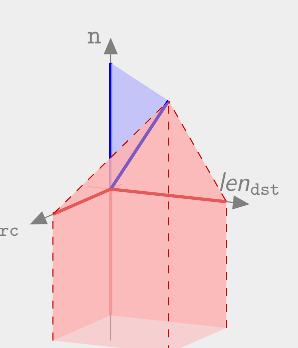
Ils permettent de représenter les propriétés avec min et max qu'on souhaite vérifier sur les programmes.

Exemple :

$$\min(\texttt{len_dst}, \texttt{n}) = \min(\texttt{len_src}, \texttt{n})$$

revient à :

$$\max(-\texttt{len_dst}, -\texttt{n}) \leq \max(-\texttt{len_src}, -\texttt{n})$$
$$\max(-\texttt{len_src}, -\texttt{n}) \leq \max(-\texttt{len_dst}, -\texttt{n})$$



Les polyèdres tropicaux (2)

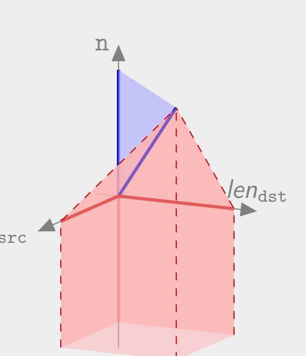
Ils permettent de représenter les propriétés avec min et max qu'on souhaite vérifier sur les programmes.

Exemple :

$$\min(\texttt{len_dst}, \texttt{n}) = \min(\texttt{len_src}, \texttt{n})$$

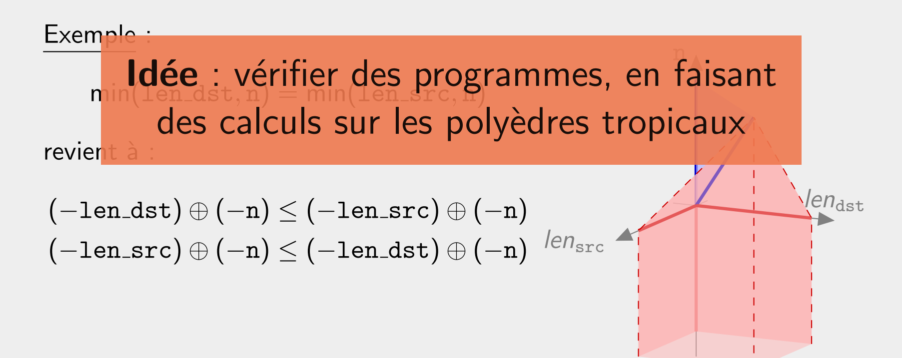
revient à :

$$(-\texttt{len_dst}) \oplus (-\texttt{n}) \leq (-\texttt{len_src}) \oplus (-\texttt{n})$$
$$(-\texttt{len_src}) \oplus (-\texttt{n}) \leq (-\texttt{len_dst}) \oplus (-\texttt{n})$$



Les polyèdres tropicaux (2)

Ils permettent de représenter les propriétés avec min et max qu'on souhaite vérifier sur les programmes.



Plan de l'exposé

Déterminer si un polyèdre est vide ...

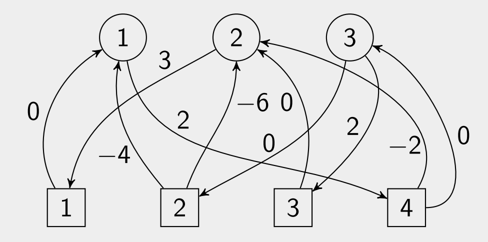
Considérons le polyèdre tropical :

$$\begin{aligned} -3 + y &\leq x \\ 0 &\leq \max(-4 + x, -6 + y) \\ -2 &\leq y \\ -2 + x &\leq \max(-2 + y, 0) \end{aligned}$$

Question : existe-t-il une solution à ces inégalités ?

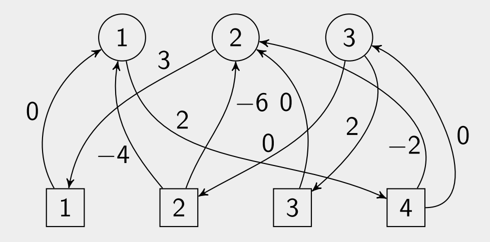
... et jouer à un jeu ...

Min et Max jouent à un jeu sur un graphe :



. . . et jouer à un jeu . . .

Min et Max jouent à un jeu sur un graphe :

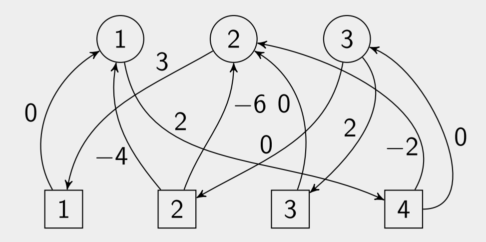


Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds

… et jouer à un jeu …

Min et Max jouent à un jeu sur un graphe :

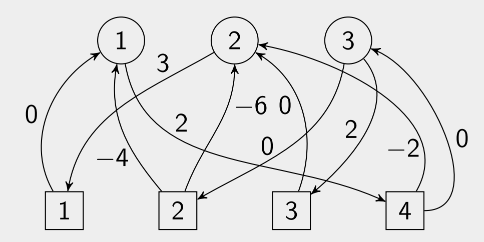


Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc

. . . et jouer à un jeu . . .

Min et Max jouent à un jeu sur un graphe :

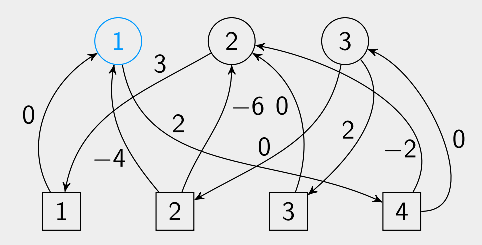


Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc
- quand le pion est sur un nœud "carré", Max choisit un arc sortant, et reçoit de Min un paiement du montant indiqué sur l'arc

... et jouer à un jeu ...

Min et Max jouent à un jeu sur un graphe :



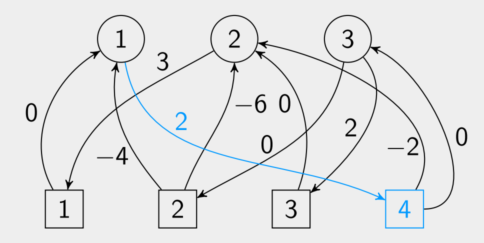
Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc
- quand le pion est sur un nœud "carré", Max choisit un arc sortant, et reçoit de Min un paiement du montant indiqué sur l'arc

En partant du nœud cercle 1 :

… et jouer à un jeu …

Min et Max jouent à un jeu sur un graphe :



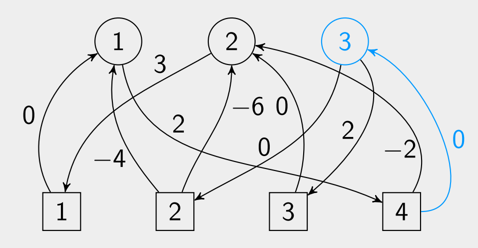
Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc
- quand le pion est sur un nœud "carré", Max choisit un arc sortant, et reçoit de Min un paiement du montant indiqué sur l'arc

En partant du nœud cercle 1 : Max gagne 2

. . . et jouer à un jeu . . .

Min et Max jouent à un jeu sur un graphe :



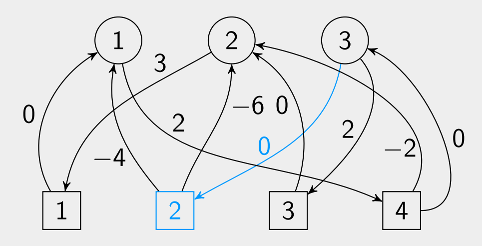
Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc
- quand le pion est sur un nœud "carré", Max choisit un arc sortant, et reçoit de Min un paiement du montant indiqué sur l'arc

En partant du nœud cercle 1 : Max gagne $2 + 0 = 2$

… et jouer à un jeu …

Min et Max jouent à un jeu sur un graphe :



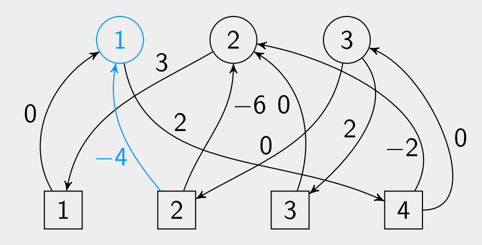
Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc
- quand le pion est sur un nœud "carré", Max choisit un arc sortant, et reçoit de Min un paiement du montant indiqué sur l'arc

En partant du nœud cercle 1 : Max gagne $2 + 0 + 0 = 2$

. . . et jouer à un jeu . . .

Min et Max jouent à un jeu sur un graphe :



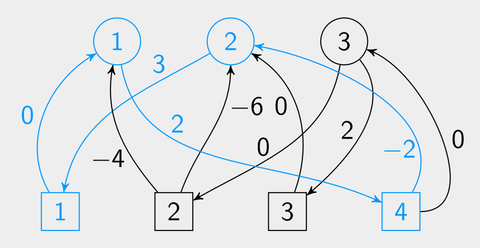
Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc
- quand le pion est sur un nœud "carré", Max choisit un arc sortant, et reçoit de Min un paiement du montant indiqué sur l'arc

En partant du nœud cercle 1 : Max gagne $2 + 0 + 0 + (-4) = -2$:-(

. . . et jouer à un jeu . . .

Min et Max jouent à un jeu sur un graphe :



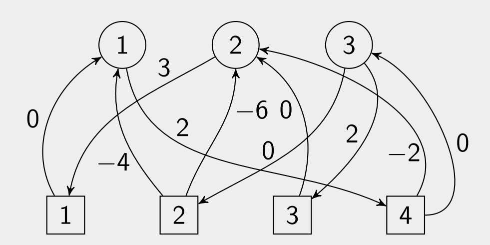
Mode d'emploi :

- les joueurs bougent alternativement un pion sur les nœuds
- quand le pion est sur un nœud "cercle", Min choisit un arc sortant, et paye à Max le montant indiqué sur l'arc
- quand le pion est sur un nœud "carré", Max choisit un arc sortant, et reçoit de Min un paiement du montant indiqué sur l'arc

En partant du nœud cercle 1 : Max gagne $2 + (-2) + 3 + 0 = 3$:-)

. . . c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

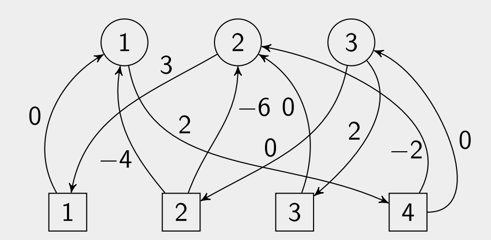


. . . c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.

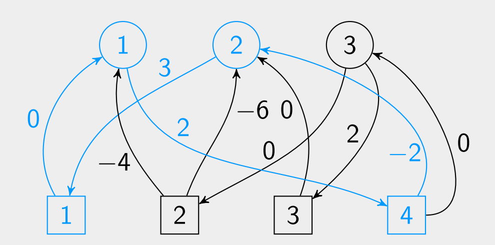


. . . c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.



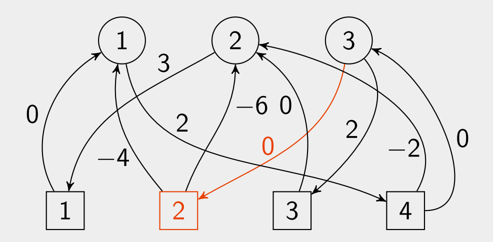
En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

… c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.



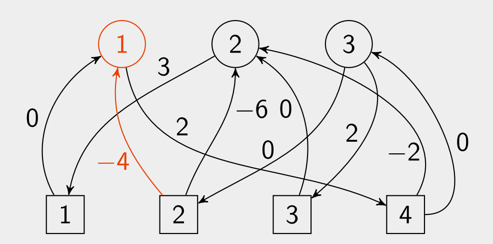
En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

. . . c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.



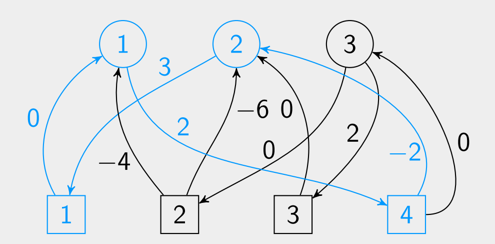
En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

... c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.



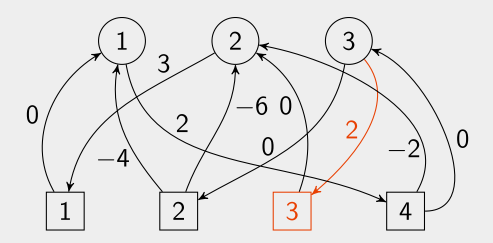
En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

... c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.



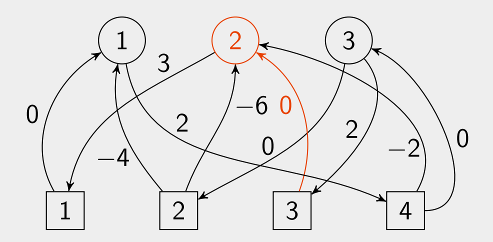
En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

... c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.



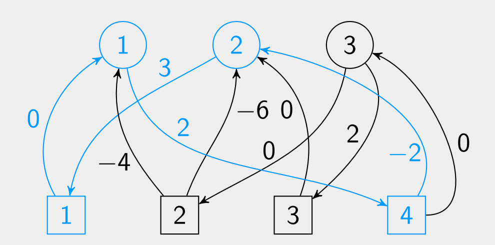
En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

... c'est la même chose

Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.



En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

. . . c'est la même chose

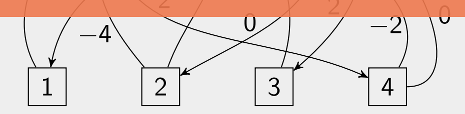
Les deux joueurs Min et Max jouent un nombre infini de fois... On regarde leur gain (ou perte) moyen par tour.

Théorème (Akian, Gaubert, Guterman)

Le polyèdre est non-vide si, et seulement si, en débutant le jeu à partir du noeud "cercle" 3, le joueur Max gagne.

Parenthèse théorique :

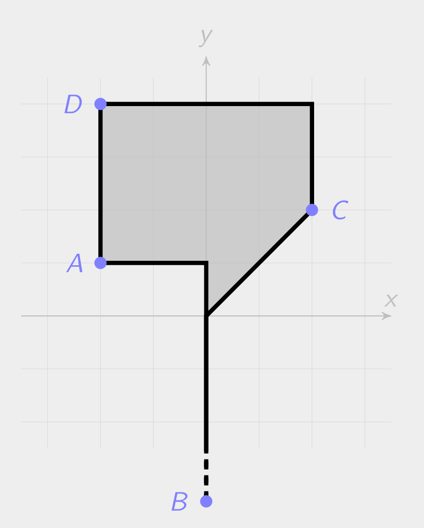
- décider si Max peut gagner est dans NP $\cap$ coNP
- l'existence d'un algorithme polynomial est un problème ouvert



En suivant le cycle en bleu, Max gagne 3$ à coup sûr.

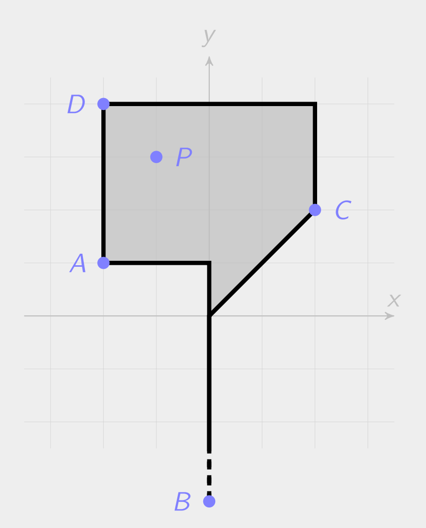
Calculer les sommets d'un polyèdre tropical

Un polyèdre tropical peut être aussi décrit par ses *sommets*.



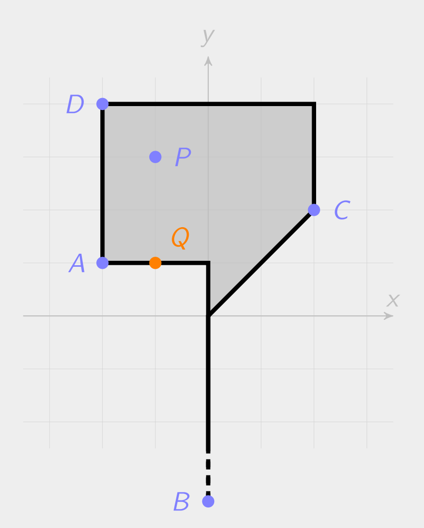
Calculer les sommets d'un polyèdre tropical

Un polyèdre tropical peut être aussi décrit par ses *sommets*.



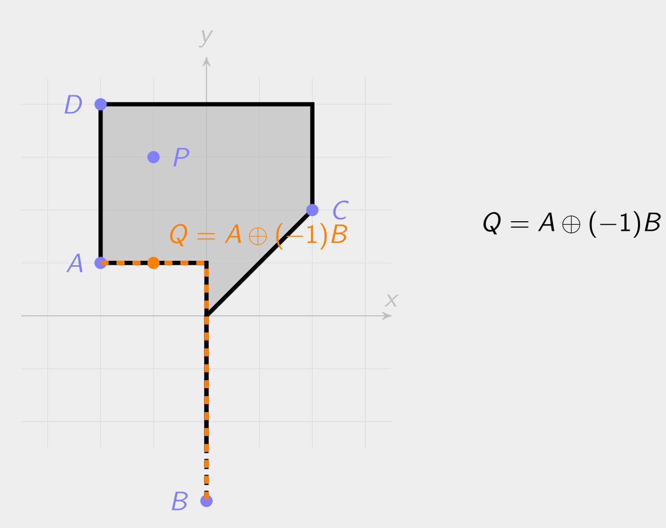
Calculer les sommets d'un polyèdre tropical

Un polyèdre tropical peut être aussi décrit par ses *sommets*.



Calculer les sommets d'un polyèdre tropical

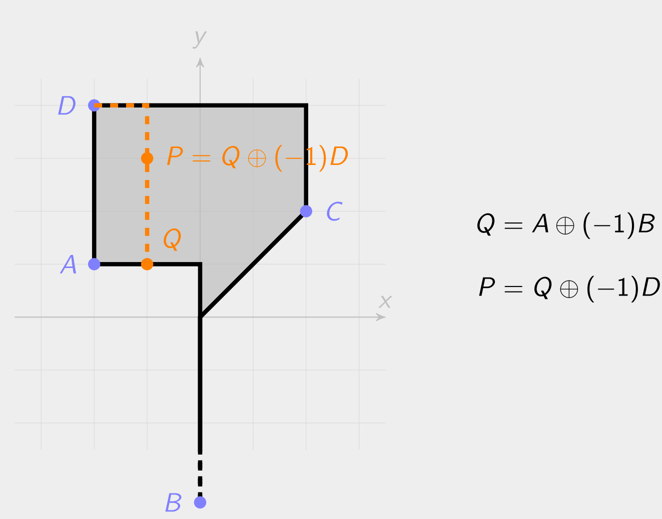
Un polyèdre tropical peut être aussi décrit par ses *sommets*.



$$Q = A \oplus (-1)B$$

Calculer les sommets d'un polyèdre tropical

Un polyèdre tropical peut être aussi décrit par ses *sommets*.

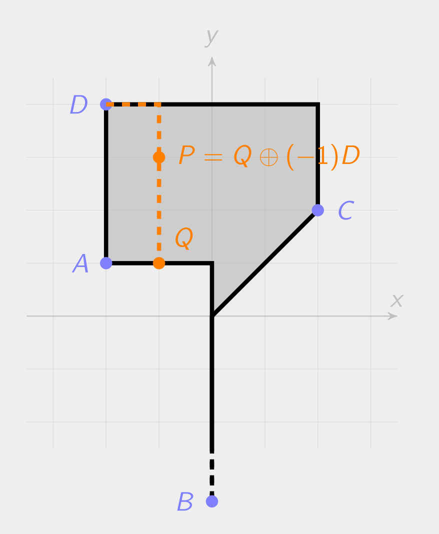


$$Q = A \oplus (-1)B$$

$$P = Q \oplus (-1)D$$

Calculer les sommets d'un polyèdre tropical

Un polyèdre tropical peut être aussi décrit par ses *sommets*.

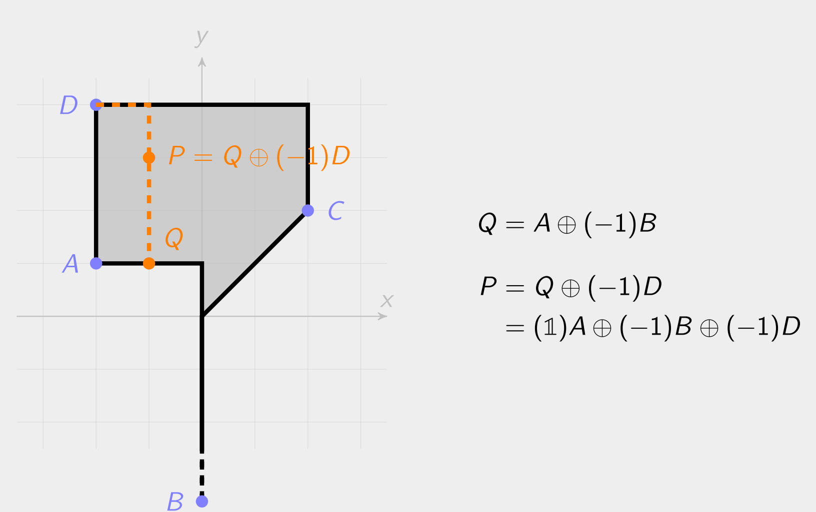


$$Q = A \oplus (-1)B$$

$$\begin{aligned} P &= Q \oplus (-1)D \\ &= (\mathbb{1})A \oplus (-1)B \oplus (-1)D \end{aligned}$$

Calculer les sommets d'un polyèdre tropical

Un polyèdre tropical peut être aussi décrit par ses *sommets*.



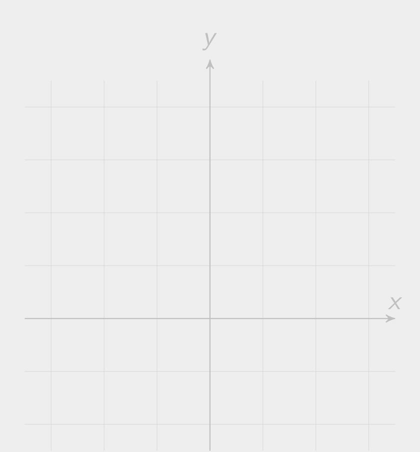
$$Q = A \oplus (-1)B$$

$$P = Q \oplus (-1)D = (\mathbb{1})A \oplus (-1)B \oplus (-1)D$$

Tout point du polyèdre peut être écrit comme un barycentre des sommets.

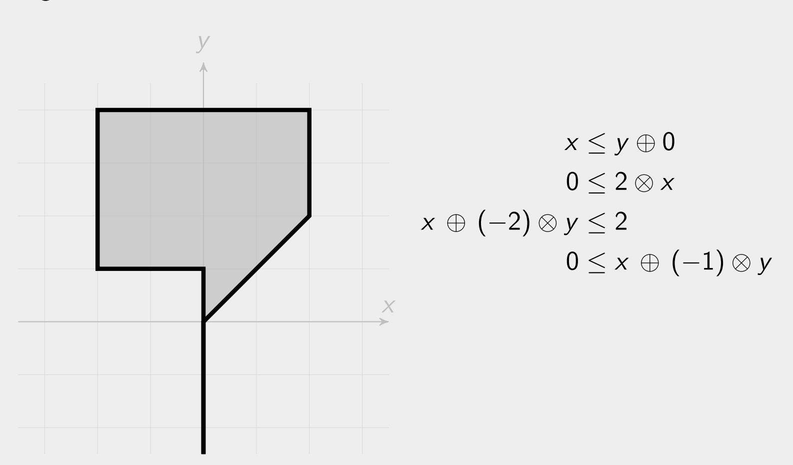
Calculer les sommets d'un polyèdre tropical (2)

Certaines opérations sont faciles à faire avec la représentation par inégalités : intersection



Calculer les sommets d'un polyèdre tropical (2)

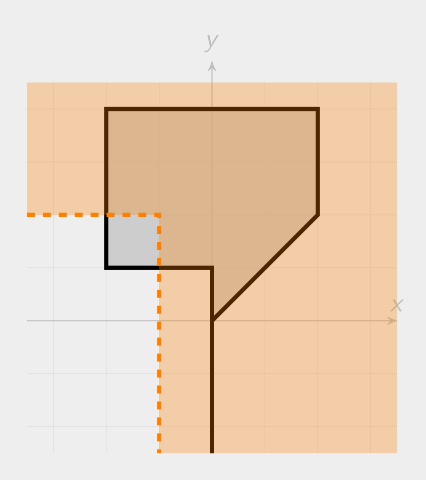
Certaines opérations sont faciles à faire avec la représentation par inégalités : intersection



$$x \leq y \oplus 0$$
$$0 \leq 2 \otimes x$$
$$x \oplus (-2) \otimes y \leq 2$$
$$0 \leq x \oplus (-1) \otimes y$$

Calculer les sommets d'un polyèdre tropical (2)

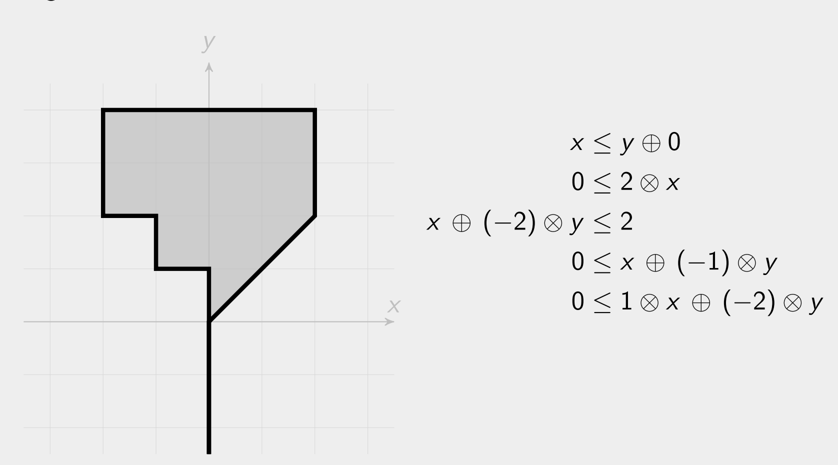
Certaines opérations sont faciles à faire avec la représentation par inégalités : intersection



$$x \leq y \oplus 0$$
$$0 \leq 2 \otimes x$$
$$x \oplus (-2) \otimes y \leq 2$$
$$0 \leq x \oplus (-1) \otimes y$$
$$0 \leq 1 \otimes x \oplus (-2) \otimes y$$

Calculer les sommets d'un polyèdre tropical (2)

Certaines opérations sont faciles à faire avec la représentation par inégalités : intersection



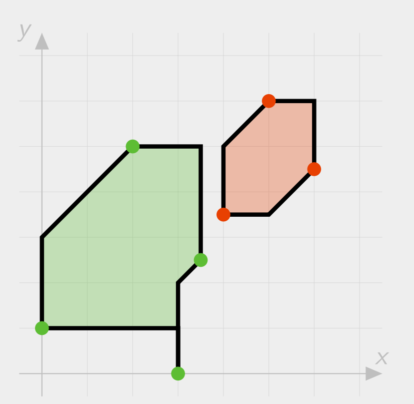
$$x \leq y \oplus 0$$
$$0 \leq 2 \otimes x$$
$$x \oplus (-2) \otimes y \leq 2$$
$$0 \leq x \oplus (-1) \otimes y$$
$$0 \leq 1 \otimes x \oplus (-2) \otimes y$$

Calculer les sommets d'un polyèdre tropical (3)

D'autres opérations sont difficiles, mais faciles sur les sommets : union (approchée)

Calculer les sommets d'un polyèdre tropical (3)

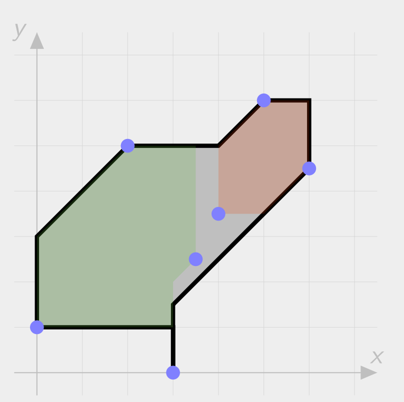
D'autres opérations sont difficiles, mais faciles sur les sommets : union (approchée)



- l'union de deux polyèdres n'est pas un polyèdre (en général)

Calculer les sommets d'un polyèdre tropical (3)

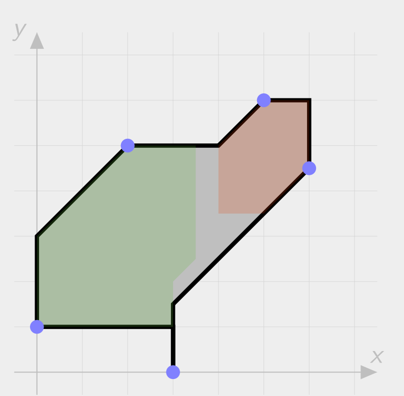
D'autres opérations sont difficiles, mais faciles sur les sommets : union (approchée)



- l'union de deux polyèdres n'est pas un polyèdre (en général)

Calculer les sommets d'un polyèdre tropical (3)

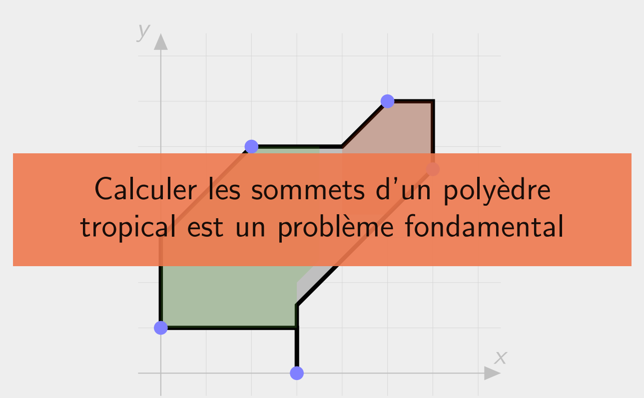
D'autres opérations sont difficiles, mais faciles sur les sommets : union (approchée)



- l'union de deux polyèdres n'est pas un polyèdre (en général)
- on sur-approxime par l'enveloppe convexe de l'union

Calculer les sommets d'un polyèdre tropical (3)

D'autres opérations sont difficiles, mais faciles sur les sommets : union (approchée)

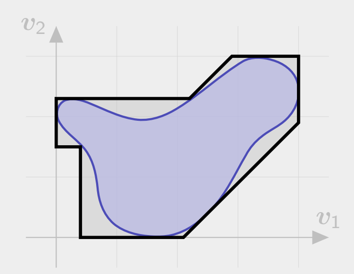


- l'union de deux polyèdres n'est pas un polyèdre (en général)
- on sur-approxime par l'enveloppe convexe de l'union

Plan de l'exposé

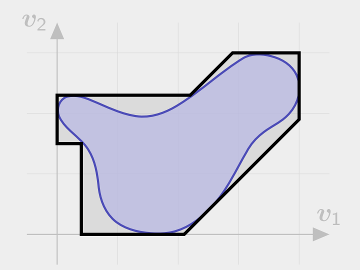
Application à la vérification

On sur-approxime la sémantique d'un programme grâce à des polyèdres tropicaux :



Application à la vérification

On sur-approxime la sémantique d'un programme grâce à des polyèdres tropicaux :

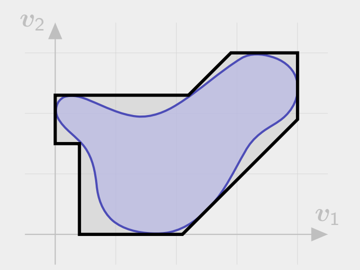


En pratique :

- la vérification de programme est réalisée grâce aux algorithmes qu'on a définis sur les polyèdres tropicaux

Application à la vérification

On sur-approxime la sémantique d'un programme grâce à des polyèdres tropicaux :

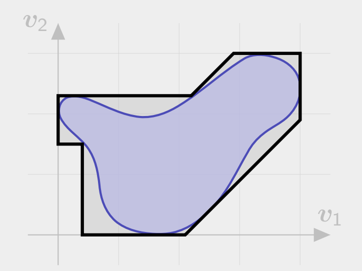


En pratique :

- la vérification de programme est réalisée grâce aux algorithmes qu'on a définis sur les polyèdres tropicaux
- c'est implémenté → TPLib, sous licence LGPL (`https://gforge.inria.fr/projects/tplib`)

Application à la vérification

On sur-approxime la sémantique d'un programme grâce à des polyèdres tropicaux :



En pratique :

- la vérification de programme est réalisée grâce aux algorithmes qu'on a définis sur les polyèdres tropicaux
- c'est implémenté → TPLib, sous licence LGPL (`https://gforge.inria.fr/projects/tplib`)
- ça fonctionne pour de vrai → **DEMO**

Plan de l'exposé

Conclusion

Contribution à la frontière entre les mathématiques et l'informatique :

- la géométrie tropicale apporte de nouvelles approches en vérification :
  - verif de programmes
  - aussi de systèmes temps-réels

- soulève aussi de très nombreuses questions théoriques
  - inventer les algorithmes en géométrie tropicale
  - mieux comprendre les propriétés de ces objets (TPLib connecté à POLYMAKE `www.polymake.org`)

Merci !